

A fast penalty-based Gauss-Seidel method for solving large-scale stochastic network constrained
unit commitment problems

By

Ananth Muruga Palani

B.E., Annamalai University, 1991
M.S., Kansas State University, 2001

AN ABSTRACT OF A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wiegers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Abstract

The exponential growth of variable renewable energy (VRE) such as wind and solar generation brings grand challenges to the operational planning of power systems. The instantaneous penetration of VRE reaches over 50% in certain balancing areas in the United States. The VRE generation is characterized by a large amount of uncertainties and variabilities. Consequently, power system operators, planners and researchers have made substantial efforts to manage VRE uncertainties in the power system scheduling, such as Network-Constrained Unit Commitment (NCUC). In order to account for the impact of VRE uncertainties, there are several noteworthy NCUC approaches in the literature, each with distinctive objectives, theories, computational requirements and economic outcomes. A common approach presented in the literature is the use of stochastic programming, namely Stochastic NCUC (S-NCUC), in which the expected system operating cost is minimized across a number of scenarios, each representing a possible realization of uncertainties. S-NCUC is typically a large-scale, non-convex, and mixed-integer programming (MIP) problem. It is modeled as a two-stage stochastic problem where the first-stage unit commitment decisions are the same for all the scenarios. Generally, S-NCUC solutions can be categorized into two main approaches. First, the most straightforward approach is to use a commercially available off-the-shelf solver to solve an extensive form (EF) of S-NCUC. However, for any large-scale system with a reasonable number of scenarios, the resulting EF of S-NCUC may become computationally intractable. To overcome this issue, the second approach is based on stage-wise or scenario-wise decomposition methods, which solve each individual scenario separately, usually in parallel, and a final solution is generated by coordinating all individual scenario solutions. Progressive Hedging Algorithm (PHA) is one of the main decomposition methods for solving the stochastic MIP. However, PHA is originally devised for a continuous convex program and is not provably convergent for the non-convex S-NCUC problem. The solution to the dual problem is generally primal infeasible and the once relaxed system-wide constraints may not be satisfied. An additional effort is required to restore the primal feasibility from a Lagrangian dual solution. Therefore, it is desirable to directly obtain a primally feasible solution from the Lagrangian dual iterations. This gives rise to exact augmented Lagrangian, a class of exact penalty methods whose objective is to solve a constrained optimization (primal) problem through an unconstrained optimization problem that has the same local (global) solutions

as the primal problem. Nevertheless, the following two critical research questions remain unresolved:

1) How can we devise an effective penalty function such that an exact solution can be obtained with a zero-duality gap?

2) If an exact solution is attained, how can we find a robust yet tight lower bound that is capable of measuring the quality of the exact solution accurately?

This dissertation addresses the aforementioned first question by applying a novel Penalty-Based Gauss-Seidel (PBGS) algorithm with an exact augmented Lagrangian representation to solve S-NCUC within a scenario-based decomposition framework. To improve the computational efficiency of PBGS, an accelerating technique that skips solving scenarios meeting certain conditions has been proposed. The proposed algorithm is named “Fast PBGS.” A proof of the Fast PBGS method is given, along with the proof of convergence of PBGS. Numerical validation of these algorithms on the IEEE 118-bus and Electric Reliability Council of Texas (ERCOT)-like large-scale systems has been carried out. Fast PBGS saves computational time by an average 35% for ERCOT-like Large System and 50% for IEEE 118-bus System with 50 scenarios compared with PBGS. Numerical results demonstrate the high quality of the PBGS solution and the efficacy of the proposed algorithms. Additionally, comparing the proposed algorithms with other prevailing S-NCUC methods such as PHA and extensive-form-based MIP solutions has been completed. The comparison of Fast PBGS shows the results are closer to EF (average difference 0.92%) than the PHA solution with EF (average difference 2.18%). When it came to the computational time, the Fast PBGS outperformed both EF and PHA. An average Fast PBGS took 48% less time than EF to obtain a solution. Compared with PHA, Fast PBGS was 142% faster.

The second question is addressed by applying the combined Frank Wolfe with PHA algorithm (FW-PHA). Our research shows that FW-PHA obtains superior lower bounds, i.e., up to 6% better than the PHA does on the IEEE 118-bus system. We further improve the computational efficiency of FW-PHA with a warm start technique that initializes the algorithm with a Fast PBGS solution. An out-of-sample analysis including a large number of samples is conducted to demonstrate the efficacy of the Fast PBGS.

A fast penalty-based Gauss-Seidel method for solving large-scale stochastic network constrained
unit commitment problems

by

Ananth Muruga Palani

B.E., Annamalai University, 1991
M.S., Kansas State University, 2001

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Mike Wiegiers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Major Professor
Dr. Hongyu Wu

Copyright

© Ananth Muruga Palani 2021.

Abstract

The exponential growth of variable renewable energy (VRE) such as wind and solar generation brings grand challenges to the operational planning of power systems. The instantaneous penetration of VRE reaches over 50% in certain balancing areas in the United States. The VRE generation is characterized by a large amount of uncertainties and variabilities. Consequently, power system operators, planners and researchers have made substantial efforts to manage VRE uncertainties in the power system scheduling, such as Network-Constrained Unit Commitment (NCUC). In order to account for the impact of VRE uncertainties, there are several noteworthy NCUC approaches in the literature, each with distinctive objectives, theories, computational requirements and economic outcomes. A common approach presented in the literature is the use of stochastic programming, namely Stochastic NCUC (S-NCUC), in which the expected system operating cost is minimized across a number of scenarios, each representing a possible realization of uncertainties. S-NCUC is typically a large-scale, non-convex, and mixed-integer programming (MIP) problem. It is modeled as a two-stage stochastic problem where the first-stage unit commitment decisions are the same for all the scenarios. Generally, S-NCUC solutions can be categorized into two main approaches. First, the most straightforward approach is to use a commercially available off-the-shelf solver to solve an extensive form (EF) of S-NCUC. However, for any large-scale system with a reasonable number of scenarios, the resulting EF of S-NCUC may become computationally intractable. To overcome this issue, the second approach is based on stage-wise or scenario-wise decomposition methods, which solve each individual scenario separately, usually in parallel, and a final solution is generated by coordinating all individual scenario solutions. Progressive Hedging Algorithm (PHA) is one of the main decomposition methods for solving the stochastic MIP. However, PHA is originally devised for a continuous convex program and is not provably convergent for the non-convex S-NCUC problem. The solution to the dual problem is generally primal infeasible and the once relaxed system-wide constraints may not be satisfied. An additional effort is required to restore the primal feasibility from a Lagrangian dual solution. Therefore, it is desirable to directly obtain a primally feasible solution from the Lagrangian dual iterations. This gives rise to exact augmented Lagrangian, a class of exact penalty methods whose objective is to solve a constrained optimization (primal) problem through an unconstrained optimization problem that has the same local (global) solutions

as the primal problem. Nevertheless, the following two critical research questions remain unresolved:

1) How can we devise an effective penalty function such that an exact solution can be obtained with a zero-duality gap?

2) If an exact solution is attained, how can we find a robust yet tight lower bound that is capable of measuring the quality of the exact solution accurately?

This dissertation addresses the aforementioned first question by applying a novel Penalty-Based Gauss-Seidel (PBGS) algorithm with an exact augmented Lagrangian representation to solve S-NCUC within a scenario-based decomposition framework. To improve the computational efficiency of PBGS, an accelerating technique that skips solving scenarios meeting certain conditions has been proposed. The proposed algorithm is named “Fast PBGS.” A proof of the Fast PBGS method is given, along with the proof of convergence of PBGS. Numerical validation of these algorithms on the IEEE 118-bus and Electric Reliability Council of Texas (ERCOT)-like large-scale systems has been carried out. Fast PBGS saves computational time by an average 35% for ERCOT-like Large System and 50% for IEEE 118-bus System with 50 scenarios compared with PBGS. Numerical results demonstrate the high quality of the PBGS solution and the efficacy of the proposed algorithms. Additionally, comparing the proposed algorithms with other prevailing S-NCUC methods such as PHA and extensive-form-based MIP solutions has been completed. The comparison of Fast PBGS shows the results are closer to EF (average difference 0.92%) than the PHA solution with EF (average difference 2.18%). When it came to the computational time, the Fast PBGS outperformed both EF and PHA. An average Fast PBGS took 48% less time than EF to obtain a solution. Compared with PHA, Fast PBGS was 142% faster.

The second question is addressed by applying the combined Frank Wolfe with PHA algorithm (FW-PHA). Our research shows that FW-PHA obtains superior lower bounds, i.e., up to 6% better than the PHA does on the IEEE 118-bus system. We further improve the computational efficiency of FW-PHA with a warm start technique that initializes the algorithm with a Fast PBGS solution. An out-of-sample analysis including a large number of samples is conducted to demonstrate the efficacy of the Fast PBGS.

Table of Contents

List of Figures	xi
List of Tables	xiv
Acknowledgements	xv
Dedication	xvi
1. Introduction.....	1
1.1. Power System Operational Planning	1
1.2. Optimization in Power System Operational Planning	3
1.2.1. Changes in the Generation Landscape	3
1.2.2. Uncertainties and Variabilities.....	5
1.3. Research	8
1.3.1. Motivation.....	8
1.3.2. Contributions.....	8
1.3.3. Organization of the Dissertation	9
1.4. Nomenclature	9
1.4.1. Operators/functions	9
1.4.2. Parameters	10
1.4.3. Variables	11
1.4.4. Abbreviations & Acronyms	11
2. Undertaking of VRE Challenges	13
2.1. Dynamic Operating Reserve	13
2.2. Chance-Constrained Programming	14
2.3. Interval Optimization	15
2.4. Robust Optimization	16
2.5. Stochastic Optimization	17
2.5.1. Stage-Wise Decomposition.....	17
2.5.2. Scenario-wise decomposition	18
2.6. Hybrid Solutions	21
2.7. Scope of The Research.....	21
3. Stochastic-Network Constrained Unit Commitment	22

3.1.	Unit Commitment (UC)	22
3.2.	S-NCUC Formulation	23
3.3.	Computational Environment	26
3.4.	Systems Modeled/Used in Simulations	29
3.4.1.	RTS-96 System	29
3.4.2.	The IEEE 118-bus System	31
3.4.3.	The ERCOT-like Large System	32
3.5.	Scenario Generation	36
3.6.	Extensive Formulation Results	46
3.6.1.	RTS-96 System	46
3.6.2.	The IEEE 118-bus	48
3.6.3.	The ERCOT-like Large System	53
3.7.	Summary	53
4.	Fast Penalty-Based Gauss-Seidel Algorithm	54
4.1.	Augmented Lagrangian	54
4.2.	Strong Duality with Augmented Lagrangian	56
4.2.1.	Affine Functions	56
4.2.2.	Non-linear Functions	57
4.3.	Exact Augmented Lagrangian	58
4.4.	Penalty-Based Gauss-Seidel Algorithm	59
4.4.1.	PBGS Algorithm	63
4.4.2.	Proof of convergence of PBGS	64
4.4.3.	PBGS Results	64
4.4.3.1.	RTS-96 System	65
4.4.3.2.	The IEEE 118-bus System	65
4.4.3.3.	The ERCOT-Like System	67
4.5.	Fast Penalty-Based Gauss-Seidel (Fast PBGS)	68
4.5.1.	Fast PBGS Algorithm	72
4.5.2.	Initialization of Implementable Z	73
4.5.3.	Comparison of PBGS vs Fast PBGS	74
4.5.3.1.	RTS-96 System	74

4.5.3.2.	The IEEE 118-bus System	77
4.5.3.3.	ERCOT-like Large System	80
4.6.	Summary	82
5.	Assessing S-NCUC Solution Quality	83
5.1.	Lower Bound and Other Methods.....	83
5.2.	Extensive Formulation (EF).....	83
5.3.	Relaxed Extensive Formulation (Re-EF).....	84
5.4.	Progressive Hedging Algorithm (PHA).....	84
5.4.1.	Comparison of Fast PBGS with PHA	89
5.4.2.	The IEEE 118-bus System	89
5.4.3.	The ERCOT-Like System.....	91
5.5.	Frank-Wolfe combined PHA (FW-PHA)	92
5.5.1.	FW-PHA Formulation	92
5.5.2.	FW-PHA Initialization.....	96
5.5.3.	FW-PHA with Warm Start.....	97
5.5.4.	Comparison FW-PHA with PHA Lower Bound	97
5.5.4.1.	RTS-96 System	97
5.5.4.2.	The IEEE 118-bus System	99
5.6.	Evaluation of Fast PBGS results.....	101
5.6.1.	RTS-96 System	101
5.6.2.	The IEEE 118-bus System	101
5.6.3.	The ERCOT-Like System.....	102
5.7.	Out-of-Sample Testing.....	103
5.8.	Discussion on Parameters	108
5.9.	Summary	109
6.	Conclusions and Future Work	110
	References	112

List of Figures

Figure 1.1 ERCOT recorded system demand on a summer day (07/18/2018).....	2
Figure 1.2 ERCOT recorded system demand on a winter day (01/15/2018).....	2
Figure 1.3 ERCOT wind generation growth 2000 – 2020 [3].....	3
Figure 1.4 Change of ERCOT generation fuel mix from 2002 to 2020 [3].....	4
Figure 1.5 SPP wind capacity and generation in SPP (left) fuel mix 2018 (right) [3]	4
Figure 1.6 US state wide installed wind capacity 1999 vs 2019 [6].....	5
Figure 1.7 Variability of renewable generation	5
Figure 1.8 Uncertainty of renewable generation.....	6
Figure 1.9 ERCOT total forecasted wind vs actual wind generation (under-forecast).....	7
Figure 1.10 ERCOT wind showing variability and uncertainty	7
Figure 2.1 Bender Decomposition showing interaction between Master and Subproblems.	18
Figure 2.2 Reliability vs computation efforts for different methods	20
Figure 3.1 NREL's FESTIV Framework	27
Figure 3.2 FESTIV - user interface.....	28
Figure 3.3 The IEEE Reliability Test System (RTS-96) diagram [74].....	30
Figure 3.4 GAMS model statistics for the RTS-96 System.....	30
Figure 3.5 The IEEE 118-bus one-line diagram	31
Figure 3.6 GAMS model statistics for the IEEE 118-bus System.....	32
Figure 3.7 Bird's-eye View of the transmission system of ERCOT.....	33
Figure 3.8 GAMS model statistics for the ERCOT-Like System.....	34
Figure 3.9 Line screening for the transmission constraints	35
Figure 3.10 The IEEE 118-bus System set 1 - load and wind, 50 Scenarios.....	38
Figure 3.11 The IEEE 118-bus System set 1 – WTG 2 and 4, 50 scenarios	38
Figure 3.12 The IEEE 118-bus System set 6 - load and wind, 50 scenarios	39
Figure 3.13 The IEEE 118-bus System set 25 – WTG 5 and 7, 50 scenarios	39
Figure 3.14 The IEEE 118-bus System set 14 - load and wind, 50 scenarios	40
Figure 3.15 The IEEE 118-bus System set 14 – WTG 3 and 10, 50 scenarios	40
Figure 3.16 The IEEE 118-bus System set 18 – load and wind, 50 scenarios.....	41

Figure 3.17 The IEEE 118-bus System set 18 – WTG 1 and 8, 50 scenarios	41
Figure 3.18 The IEEE 118-bus System set 25 - load and wind, 50 scenarios	42
Figure 3.19 The IEEE 118-bus System set 25 – WTG 6 and 9, 50 scenarios	42
Figure 3.20 The ERCOT-like System – load and wind, 30 scenarios	43
Figure 3.21 The ERCOT-like System – wind, 30 scenarios.....	43
Figure 3.22 The ERCOT-like System – WTG 15 and 25, 30 scenarios.....	44
Figure 3.23 The ERCOT-like System – WTG 63 and 108, 30 scenarios.....	44
Figure 3.24 The ERCOT-like System total load variation across scenarios for each hour	45
Figure 3.25 Partial output of EF CPLEX solution for RTS96-10-S1	47
Figure 3.26 FESTIV output of EF solution for scenario RTS96-10-S1	48
Figure 3.27 Partial output of EF CPLEX solution for IEEE118-10-S1.....	49
Figure 3.28 FESTIV output of EF solution for scenario IEEE118-10-S1	50
Figure 3.29 Partial output of EF CPLEX solution for IEEE118-50-S0.....	51
Figure 3.30 FESTIV output of EF solution for scenario IEEE118-50-S0.....	52
Figure 4.1 Duality gap in minimization problems lacking adequate convexity [87].....	56
Figure 4.2 Duality gap removed by an augmenting function [87].....	57
Figure 4.3 Value function and some augmenting functions [90].....	58
Figure 4.4 Number of scenarios solved in each iteration for the RTS-96 System using Fast PBGS	76
Figure 4.5 Number of scenarios solved in each iteration for the IEEE 118-bus System using Fast PBGS.....	78
Figure 4.6 Number of scenarios solved in each iteration for the IEEE 118-bus System with 50 Scenarios	79
Figure 4.7 Number of scenarios solved in each iteration for the ERCOT-like large System using Fast PBGS	81
Figure 5.1 Comparison of Fast PBGS with PHA for the IEEE 118-bus System.....	90
Figure 5.2 Comparison of Fast PBGS with PHA for the ERCOT-like System.....	92
Figure 5.3 Flowchart of combined Frank-Wolfe and PHA	95
Figure 5.4 Initialization of convex hull for FW-PHA.....	96
Figure 5.5 Lower bound using normal start vs warm-start for the IEEE 118-bus System	97
Figure 5.6 Sensitivity of lower bound to ρ for RTS-96 System	98

Figure 5.7 Lower bound over iterations for RTS-96 System.....	98
Figure 5.8 Sensitivity of lower bound to ρ for RTS-96 System	99
Figure 5.9 Lower bound over iterations for the IEEE 118-bus System (IEEE118-10-S1).....	100
Figure 5.10 Lower bound over iterations for the IEEE 118-bus System (IEEE118-50-S0).....	100
Figure 5.11 Comparison of objective values for the IEEE 118-bus system	105
Figure 5.12 Comparison of computational performance for the IEEE 118-bus system.....	107

List of Tables

Table 3.1 List of scenarios sets with scenario generation statistics	37
Table 4.1 Comparison of PBGS with EF - set RTS96-10-S1	65
Table 4.2 Comparison of PBGS with EF - set IEEE118-10-S1	66
Table 4.3 Comparison of PBGS with EF - set IEEE118-50-S0	67
Table 4.4 PBGS solution for the ERCOT-Like System, set ERCOT-30-S1	67
Table 4.5 Comparison of PBGS with Fast PBGS - set RTS96-10-S1	74
Table 4.6 Comparison of PBGS with Fast PBGS - set IEEE118-10-S1	77
Table 4.7 Comparison of PBGS with Fast PBGS - set IEEE118-50-S0	77
Table 4.8 Comparison of PBGS with the Fast PBGS - set ERCOT-30-S1	80
Table 5.1 Comparison of Fast PBGS with Re-EF	84
Table 5.2 Comparison of Fast PBGS with PHA for the IEEE 118-bus System.....	90
Table 5.3 Assessment of Fast PBGS result using FW-PHA LB for RTS-96 System	101
Table 5.4 Assessment of Fast PBGS result using FW-PHA LB for the IEEE118-bus System..	102
Table 5.5 Assessment of Fast PBGS result using FW-PHA LB for the ERCOT-like System...	102
Table 5.6 Comparison of Fast PBGS objective value with EF and PHA for the IEEE 118-bus System.....	104
Table 5.7 Comparison of Fast PBGS computational time with EF and PHA for the IEEE 118-bus System.....	106

Acknowledgements

I want to express my deepest gratitude to my advisors Dr. Medhat M. Morcos and Dr. Hongyu Wu. I could not have started and stayed on the program without continuous encouragement and guidance from Dr. Morcos. I could not have realized my dream of completing the doctoral program without Dr. Wu's excellent guidance and timely help. I am blessed to have both of them guiding me. I owe a lot to both.

My sincere thanks to committee members Dr. William Hsu and Dr. Hani Melhem for their precious time. I want to thank Dr. Bruce Law for his time as the outside chairperson and Dr. Andrew Rys, Graduate program director, for helping me in restarting the program.

I want to reflect upon the sacrifices made by my parents R. Palani Mudaliyar and Kalaiselvi Palani, three decades ago when they put my education as the highest priority of their life. I also would like to reflect upon my grandfather A. Kannyappa Mudaliyar who raised me and instilled good character into me. Though all three have left me to be with Siva (God) in the last few months, I am sure they continue to bless me from there.

When it comes to sacrifices, I cannot quantify what my wife Priya did during the program. I am deeply thankful to her. Also, I would like to thank my son Paari, daughter Kavya and my in-laws Mr. and Mrs. A.K. Kannan, for their encouragement. I appreciate my brother Pravin and his family's support. I am grateful to Mrs. Sharon Morcos for her prayers for me during difficult times. In addition, I am thankful to Mrs. Shailaja Atteri who was so interested in seeing me complete the program.

I want to thank Dr. Juan Santos, my colleague, for helping me with models, proofreading, and providing constant encouragement. Also, my sincere thanks to Dr. Erik Ela of EPRI, Dr. James Luedtke of the University of Wisconsin, Dr. Mohammad Feizollahi of Georgia State University, and Dr. Fabricio Oliveira of Aalto University for fruitful discussion and feedback.

Last but not least, I would like to express my appreciation to the researchers, at the National Renewable Energy Laboratory in Golden, Colorado. The Flexible Energy Scheduling Tool for Integrating Variable generation (FESTIV) provided the jump start I needed to complete my research.

Dedication

I dedicate this dissertation to the loving memory of my parents R. Palani Mudaliyar and Kalaiselvi Palani. Without their love, hard work, and selflessness, I would not be who I am today.

1. Introduction

This chapter introduces the contents of the dissertation. Section 1.1 briefly describes the Power System Operational Planning (PSOP). The research is conducted in the context of PSOP. In section 1.2, challenges faced in PSOP due to the changing generation landscape are presented. This challenge is the reason for the motivation of this research. Finally, in section 1.3, the contribution made is given.

1.1. Power System Operational Planning

The PSOP goal is to fundamentally supply electric energy to the customers in the most economically and reliably fashion. This planning endeavor by the utility/Independent System Operators (ISO) usually starts a week before the operating day and continues through the hour ahead of the real-time operations. It is well known that the demand for electrical energy must be met with supply in real-time, unlike other commodities. This requirement is complicated because the resources that supply reliable energy have temporal constraints such as minimum generator start time. These constraints force a decision to be made well in advance of real-time by looking at the forecasted demand. Actual energy demand varies by time of the day and season of the year. Figures 1.1 and 1.2 show these wide variations for the Electric Reliability Council of Texas (ERCOT) ISO. The forecast of demand itself is a challenge for an extensive geographically-spanning system with varying weather across the geography. Economic decision-making on scheduling generators to meet the demand is exponentially complex for increasing the number of generators, each with a dozen constraints. The reliability of the system requires respecting the transmission network constraints, which further exacerbates the complexity.

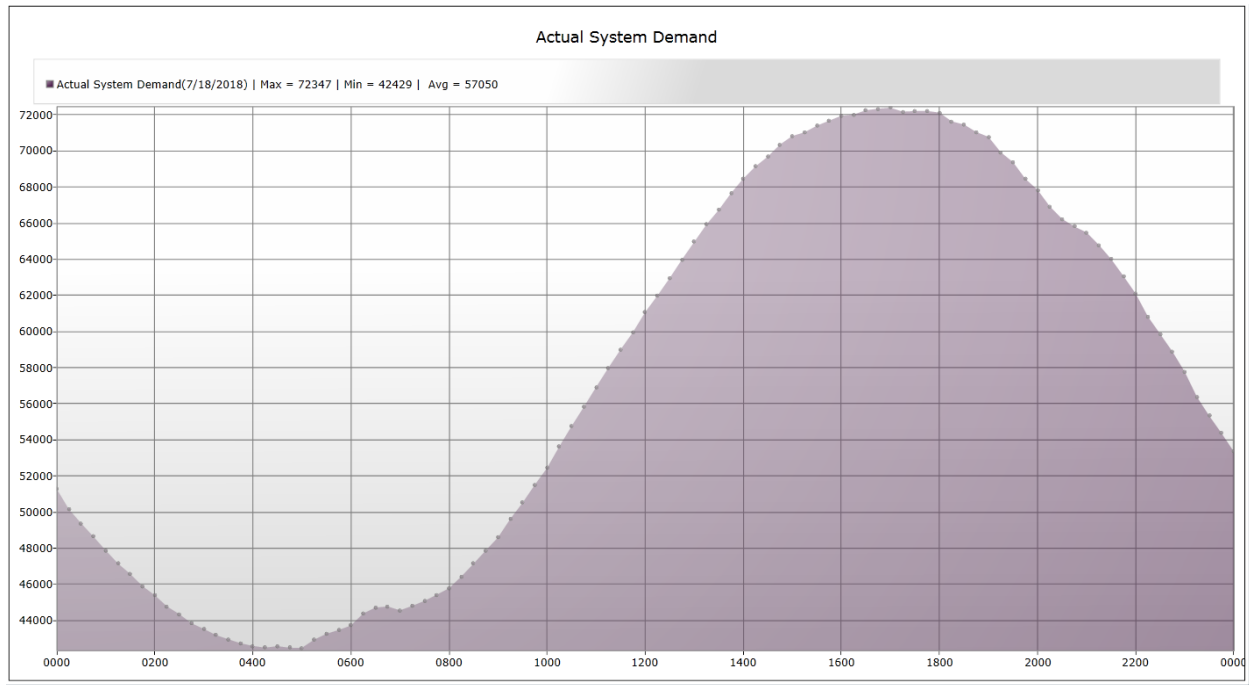


Figure 1.1 ERCOT recorded system demand on a summer day (07/18/2018)

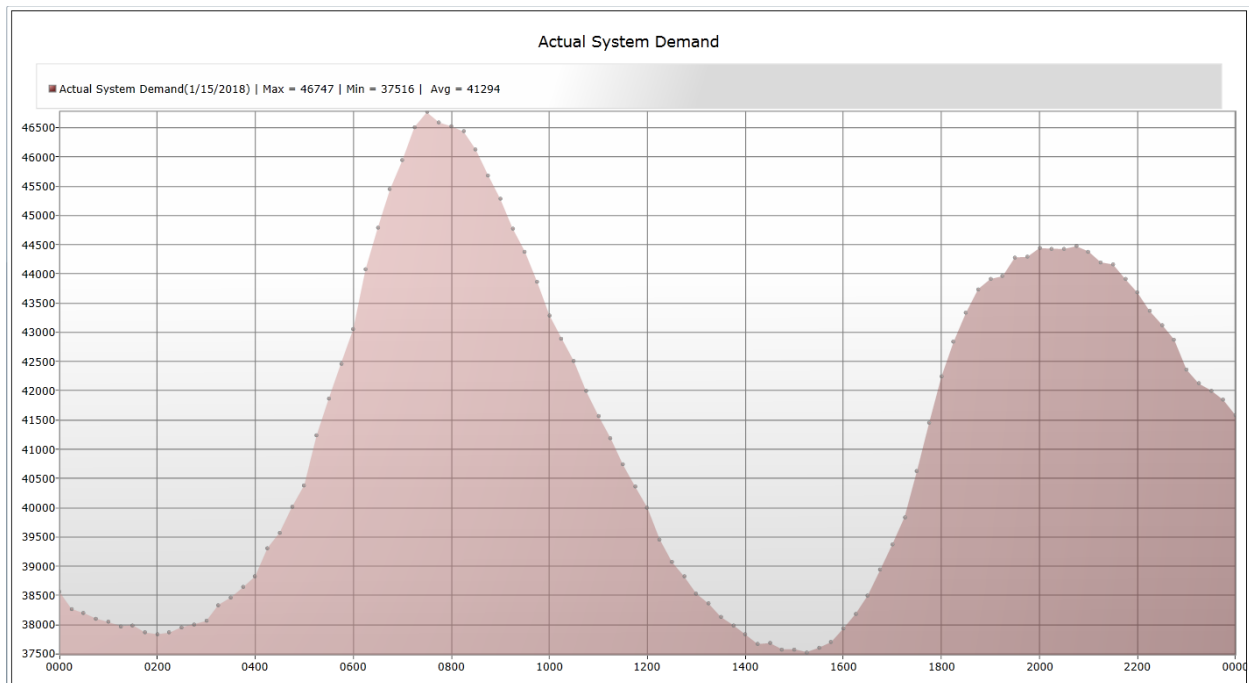


Figure 1.2 ERCOT recorded system demand on a winter day (01/15/2018)

1.2. Optimization in Power System Operational Planning

The earliest literature references the economic operation of power system operations date back to the early 1940s [1]. PSOP uses an application called Security Constrained Unit Commitment (SCUC). The SCUC is an optimization problem to minimize the cost of generation unit commitment and schedule needed to meet the forecasted demand subject to a host of constraints. A detailed formulation of this optimization problem is given in Chapter 3. Mixed Integer Linear Programming (MILP) application in scheduling thermal generating systems dates back to 1968 [2]. Until now, this SCUC has been working well for the entities in PSOP. The only unknown is electricity demand, and utilities have successfully been forecasting this within 2 to 3% error.

1.2.1. Changes in the Generation Landscape

Wind energy started making into utility-scale generation at the beginning of the millennium. As shown in Figure 1.3, wind energy in the year 2000 was just over 100 MW grew into about 21000 MW in 2017. This wind capacity was projected to be close to 30,000 MW by year 2020 in the report released by ERCOT in 2017 [3]. Figure 1.4 shows this exponential growth. The wind has taken over the place of the coal units as the second most generation type based on the fuel. It has replaced a small share of the combined cycle (CC) generation too. Similar exponential growth has started showing up in solar power generation. Today the ERCOT region has 1,900 MW of solar power generation, and nearly 60,000 MW of solar power generation is under study.

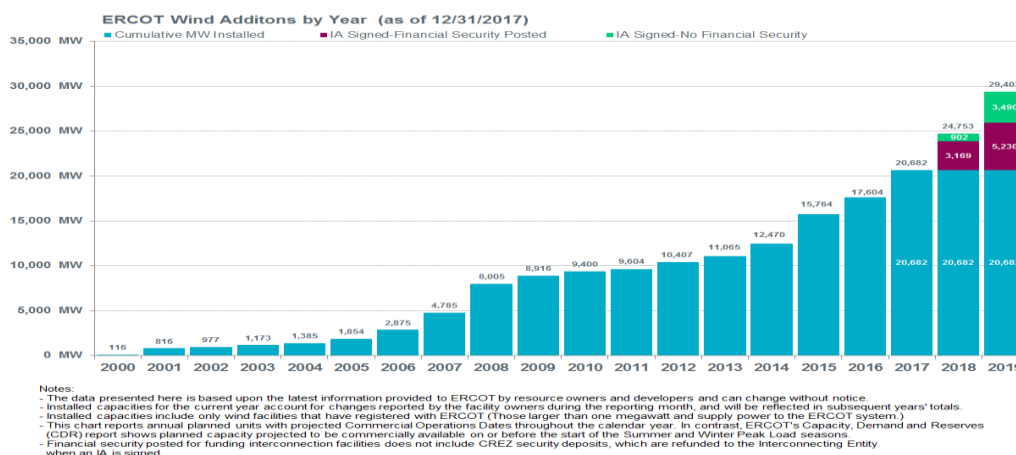


Figure 1.3 ERCOT wind generation growth 2000 – 2020 [3]

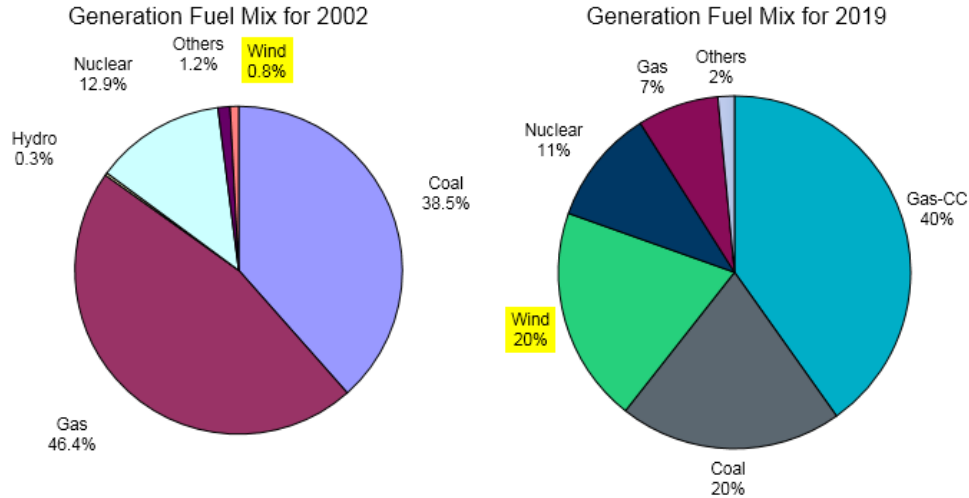


Figure 1.4 Change of ERCOT generation fuel mix from 2002 to 2020 [3]

Southwest Power Pool (SPP) region has seen similar wind generation growth over the same period. Figure 1.5 shows wind capacity and generation growth over 12 years, along with the generation fuel mix at the end of 2018 [4], with the wind at 23%. In California ISO (CAISO), about 27% of its demand is supplied by renewables [5]. This renewable growth is not unique to ERCOT, SPP, and CAISO. It is seen in other regions in the US and around the world. Figure 1.6 shows installed the wind capacity in 1999 vs. 2019 [6].

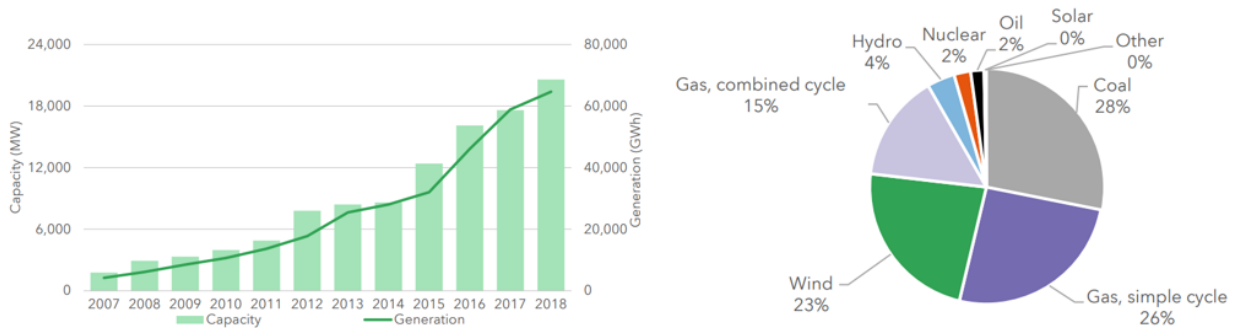


Figure 1.5 SPP wind capacity and generation in SPP (left) fuel mix 2018 (right) [3]

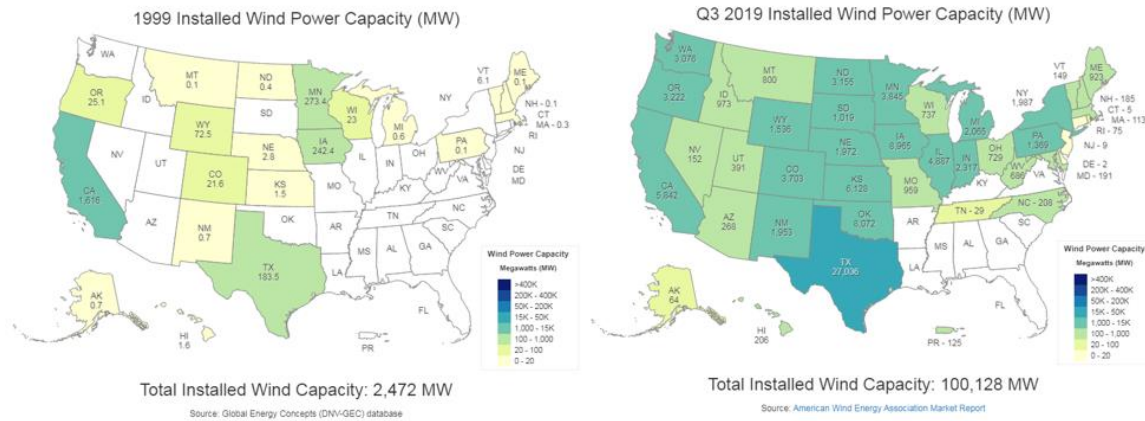


Figure 1.6 US state wide installed wind capacity 1999 vs 2019 [6]

1.2.2. Uncertainties and Variabilities

It is beneficial that this environmentally friendly, zero fuel-cost generation is replacing high emission, \$20-\$30/MWh fuel cost, thermal (mostly coal) units. On the negative side, the renewable generation outputs are not fixed like thermal generation capacity. The actual generation depends on wind speed or solar irradiation, which are not easy to predict, especially wind speed. The generation output cannot be kept at a constant value. In other words, they have variability and uncertainty, and they are not as reliable as thermal generation.

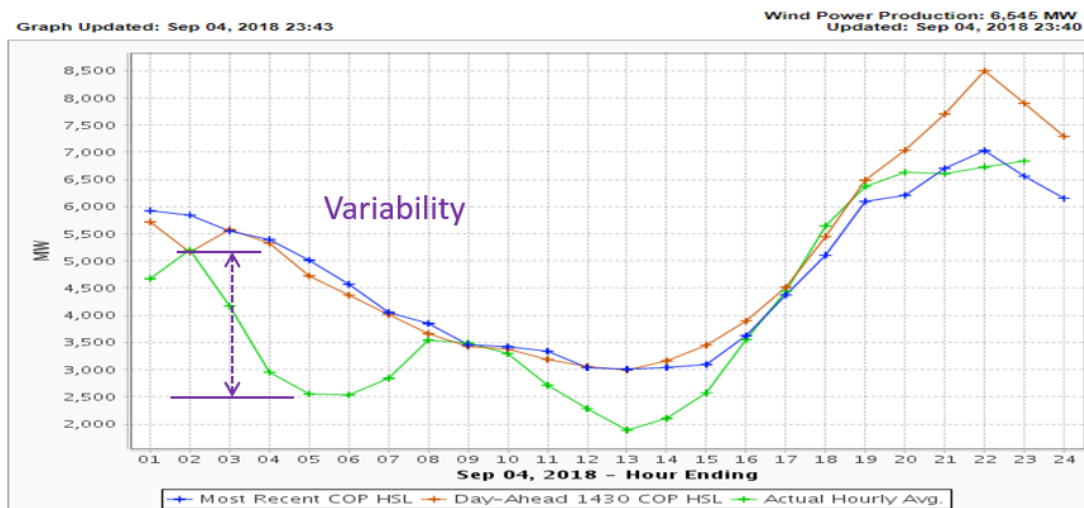


Figure 1.7 Variability of renewable generation¹

¹ www.ercot.com

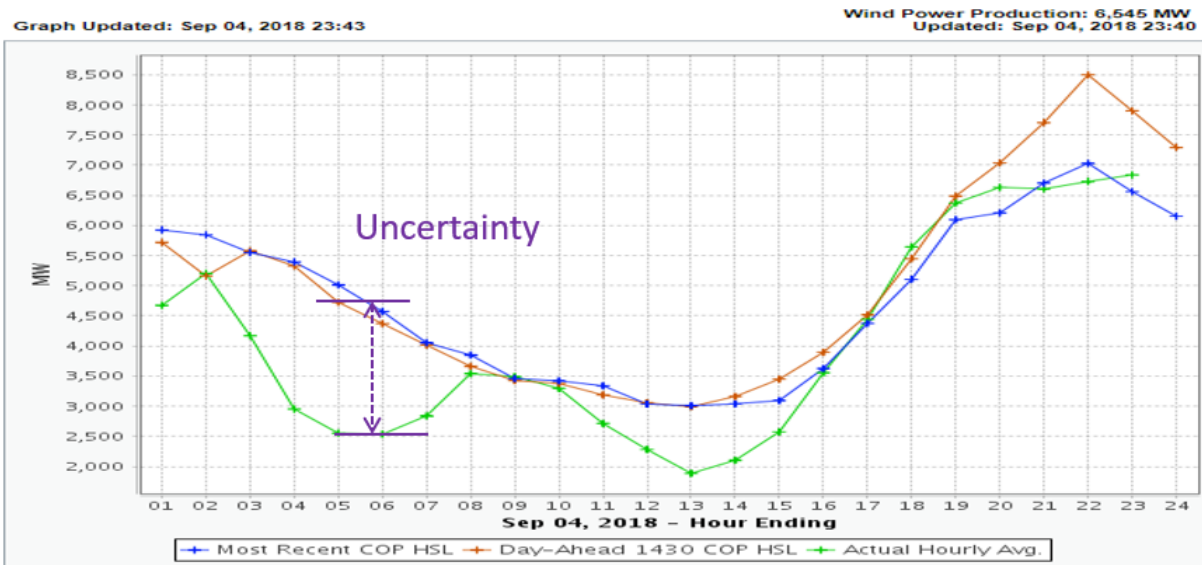


Figure 1.8 Uncertainty of renewable generation¹

Figures 1.7 and 1.8 show ERCOT forecast in day-ahead for September 4, 2018, and even the hour-ahead forecast predicted was 2500 MW higher than what actual production turned out to be for the early morning hours. While Figure 1.8 shows the over-forecast, Figure 1.9 shows under-forecast. On August 31, 2016, the actual generation came out to be as high as 3000 MW more than both the day-ahead and hour-ahead forecasts. These two plots underscore the uncertainty in forecasting wind generation. They also show the variability as the wind generation varies between the hours. The variability is much more drastic on March 19, 2020, as shown in Figure 1.10. One can observe the variability and uncertainty for the ERCOT by visiting www.ercot.com.

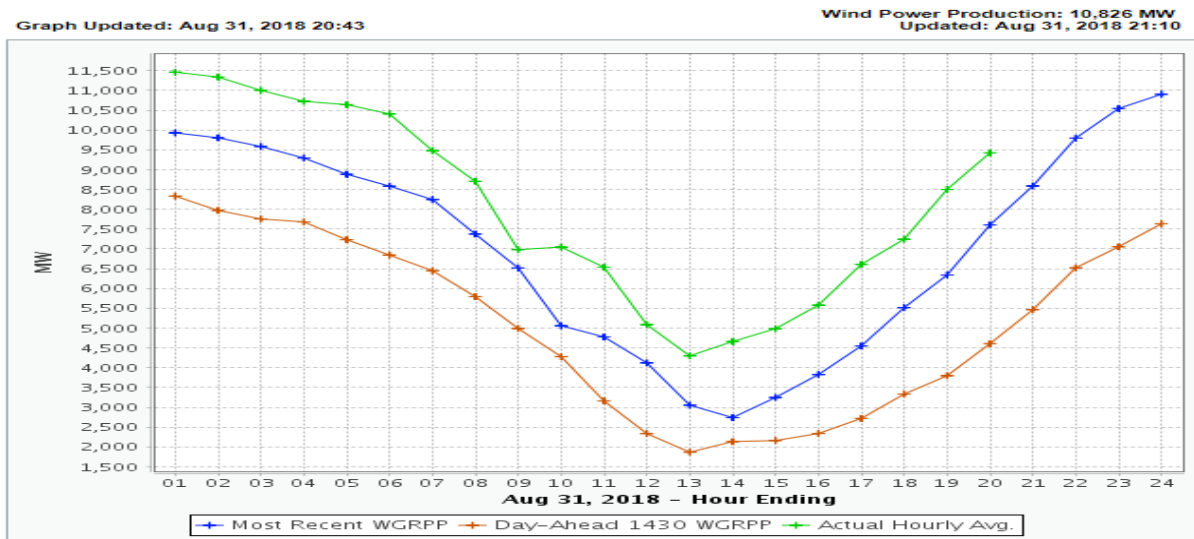


Figure 1.9 ERCOT total forecasted wind vs actual wind generation (under-forecast)¹

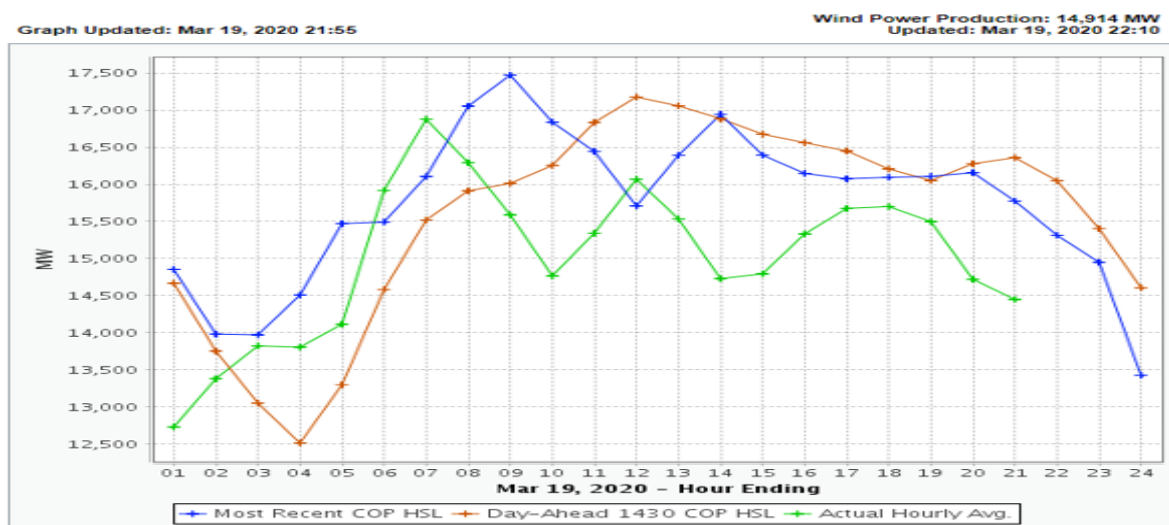


Figure 1.10 ERCOT wind showing variability and uncertainty¹

1.3. Research

1.3.1. Motivation

This changing generation mix creates a challenge in PSOP. It is not just that the VRE creates uncertainties; the demand side management participation in the organized markets is causing additional uncertainties. The current SCUC optimization application is not designed to handle parameters that are not deterministic. A new algorithm that can optimize under uncertainty must be applied in solving SCUC. The research is motivated by the challenge brought in by the exponential growth of VRE and the necessity to find a solution. Penalty-Based Gauss-Seidel algorithm (PBGS), which solves stochastic SCUC and yields a primal feasible solution when it converges, has been found in this research. An efficient way of speeding up the solution and measuring such a solution's efficacy has been developed.

1.3.2. Contributions

Here are the contributions to this PSOP through the research.

- (1) The PBGS algorithm is applied for the first time, including a positive-basis exact penalty representation and an implementable binary calculation, to produce an exact solution to S-NCUC. The exactness guarantees the primal feasibility of a Lagrangian dual solution and thus eliminates the need for extra algorithms to restore the primal feasibility at the cost of optimality.
- (2) Fast PBGS algorithm has been developed by improvising PBGS that reduces the solution time by 15-40% compared with the PBGS algorithm. A mathematical proof of Fast PBGS has been included.
- (3) The use of a PBGS solution is proposed to initialize the FW-PHA algorithm over traditional initialization. The proposed initialization yields an improved lower bound that allows for measuring the quality of an exact solution more precisely and efficiently.
- (4) The proposed methodology is validated and compared with other algorithms, especially on an ERCOT-like, large-scale system. The system consists of 7226 buses, 8853 branches, and 725 generators, including 178 wind generators. This system represents the most

extensive realistic system in S-NCUC studies. This is the first time S-NCUC is solved for such a sizeable real-world system.

Contributions (1), (2) and (4) are discussed in Chapter 4 and in the following article:

A. M. Palani, H. Wu and M. M. Morcos, "A fast penalty-based Gauss-Seidel method for stochastic unit commitment with uncertain load and wind generation," in *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 211-222, 2021.

Contribution (3) is discussed in Chapter 5 and in the following article:

A. M. Palani, H. Wu, and M. M. Morcos, "A Frank–Wolfe progressive hedging algorithm for improved lower bounds in stochastic SCUC," *IEEE Access*, vol. 7, pp. 99398–99406, 2019.

1.3.3. Organization of the Dissertation

A literature review of research and development in stochastic programming applied in solving S-NCUC is discussed in Chapter 2. Chapter 3 presents the development of S-NCUC using EF, where it is shown that the computational time of EF increases exponentially with an increased number of scenarios. In Chapter 4, scenarios-wise decomposition algorithm, PBGS is applied to solve S-NCUC. In the same chapter, an essential contribution by the research, the Fast PBGS, is devised, and the computational comparison is made between PBGS and the Fast PBGS. In Chapter 5, a lower bound method, with an improvised FW-PHA, is applied to obtain a solution to assess the Fast PBGS solution. Finally, in Chapter 6 conclusion remark is made along with future research opportunities is discussed.

1.4. Nomenclature

1.4.1. Operators/functions

$\Psi_\rho(\cdot)$	<i>Penalty function used in Augmented Lagrangian</i>
$[\cdot]$	<i>Rounding Operator.</i>
$[\cdot]^-$	$-\min\{0, \cdot\}$
LR	<i>Lagrangian relaxation</i>
$LR +$	<i>Augmented Lagrangian</i>

1.4.2. Parameters

NT	Number of time periods
N	Number of load buses
NI	Number of generation units
S	Number of scenarios
NG	Number of segments in production cost curve
L	Number of transmission branches
t	Index for time periods
i	Index for units: 1,2, \dots , NI
s	Index for scenarios: 1,2, \dots , S
d	Index for cost curve segments: 1,2, \dots , NG
Pr^s	Probability of scenario s
NL_i	No load cost of unit i [\$/h]
$IC_{d,i}$	Incremental cost of unit i , seg d [\$/MW]
β	PBGS convergence acceleration parameters
γ	PBGS Step size
$\underline{\rho}, \underline{\rho}, \bar{\rho}$	Penalty factors, initially $\underline{\rho} = \bar{\rho} = \rho$
ϵ	PBGS tolerance limit for $\Delta\phi$
ξ	Probabilly – used in Chance Constrained Optimization
$\overline{Pg}_i / \underline{Pg}_i$	Max/ min power output of unit i [MW]
RU_i / RD_i	Unit i ramp up/down limit [MW/h]
$SU_{i,t}$	Startup cost of unit i at time t in scenario s [\$]
$D_{n,t}^s$	Demand at bus n at time t scenario s [MW]
TU_i / TD_i	Unit i min. up/down time [h]
$\overline{LF} / \underline{LF}$	Branch flow limit [MW]
$VOLL$	Value of lost load [\$/MWh]
$VOOB$	Value of overloaded branch [\$/MWh]

1.4.3. Variables

$I_{i,t}^s$	State of unit i at time t in scenario s , 1 for <i>ON</i> and 0 for <i>OFF</i>
$IU_{i,t}^s$	Startup indicator of unit i at time t in scenario s
$ID_{i,t}^s$	Shutdown indicator of unit i at time t in scenario s
$P_{d,i,t}^s$	Dispatch of unit i at segment d at time t in scenario s [MW]
$Pg_{i,t}^s$	Dispatch of unit i at time t in scenario s [MW]
Φ	Lower bound
ϕ	Lower bound
$D_{n,t}^s$	Demand at bus n at time t scenario s [MW]
LL_t^s	Loss of load at time t in scenario s [MW]
AL_t^s	Additional load at time t in scenario s [MW]
$Z_{i,t}$	Implementable state of unit i at time t
$LF_{l,t}^s$	Line flow at time in t in scenario s [MW]
$BrSl1_{l,t}^s$	Branch Slack1 at time t in scenario s [MW]
$BrSl2_{l,t}^s$	Branch Slack2 at time t in scenario s [MW]
$\omega_{i,t}^s$	Lagrangian multiplier (scaled by scenario probability)
λ	Lagrangian multiplier
Λ^s	Feasible region for NCUC for a scenario s
V_s	Convex hull of feasible region of NCUC for a scenario $s = Conv(\Lambda^s)$

1.4.4. Abbreviations & Acronyms

ALD	Augmented Lagrangian Dual
AS	Ancillary Services
CAISO	California ISO
ED	Economic Dispatch
ERCOT	Electric Reliability Council of Texas
F-PBGS	Fast PBGS
FW	Frank-Wolfe Algorithm
ISO	Independent System Operator
LB	Lower Bound
LD	Lagrangian Dual

LP	Linear Programming
LR	Lagrangian Relaxation
MILP	Mixed-Integer Linear Programming
MIP	Mixed-Integer Programming
NAC	Non-anticipativity Constraint
NCED	Network Constrained Economic Dispatch
NCUC	Network Constrained Unit Commitment
PBGS	Penalty-Based Gauss-Seidel
PHA	Progressive Hedging Algorithm
PSOP	Power System Operation and Planning
QSG	Quick Start Generation
RO	Robust Optimization
RTC	Real-time commitment
RTO	Reginal Transmission Organization
RUC	Reliability Unit Commitment
SCED	Security Constrained Economic Dispatch
SCUC	Security Constrained Unit Commitment
SO	Stochastic Optimization
SPP	Southwest Power Pool
TSO	Transmission System Operators
UC	Unit Commitment
VRE	Variable Renewable Energy

2. Undertaking of VRE Challenges

System operators such as ISOs and regulated utilities use a critical application called Unit Commitment (UC) in power system operations. UC is an optimization problem which is typically run in Day-Ahead (DA) to plan the following day operation (the operating day). The objective of UC is to minimize the cost of generation commitment and production while enforcing the demand requirements (load balancing) and host of other constraints such as transmission limits. A detailed formulation of UC is presented in Chapter 3.

Parameters used in solving UC problems have been deterministic until recently. The growth of VRE has brought uncertainties to these parameters used in UC problems. Several noteworthy UC approaches can be found in the literature [7]-[8], each with different objectives, mechanisms, computational requirements, and economic outcomes that account for the uncertainties in the UC parameters. The solutions in the current research that tackle the VRE challenges are discussed here.

2.1. Dynamic Operating Reserve

System operators have historically maintained system reserves such as online (spinning) and offline (non-spinning) reserves to meet unexpected generation or load deviation from the plan or forecast due to the forecast's error. In general, reserves are maintained to meet any unforeseen situation that requires additional generation. The development of markets created more categories of these reserves and are often called ancillary services. There are different types of ancillary services often referred to as commodities with varying requirements for specific needs. For example, regulation reserves procurement to keep the frequency close to the scheduled value is different from a reserve used when a forced outage of a generation unit occurs. In any case, some types of reserves based on loss of the largest generation unit have a fixed value for all the operating hours, and the other reserves amount procured are based on statistical and probability analysis. One of the earliest applications of such analysis dates back to the early 1960s [9]. Even today, ERCOT ancillary services are determined based on the statistical analysis of the historical information and the probability of future system conditions [10].

In this approach, the operating reserve capacity is determined on an hourly basis, or block of hours, based on expected renewable production and the historical impact of wind forecast errors. Matos and Bessa [11] proposed a new reserve management tool based on probabilistic wind power forecasts to determine operating reserve needs. Holttinen *et al.* compared methods used in wind integration analyses and operating practice [12]. Through the research of different operational practice methods due to wind integration, the authors found that wind variability is not a contingency event. Instead, the impact of wind generation on the reserves was a nonevent operation. Also, the authors determined that some events of more considerable variability and more significant forecast errors could be categorized as slow events. Therefore, the level of operating reserve needed for wind is not constant during all hours of the day, and the dynamic allocation of reserves would be more efficient. Computation of dynamic operating balancing reserve for wind power integration for hours 1 to 48 of operation was presented by Menemenlis *et al.* in [13]. De Vos has conducted research on sizing and allocating operating reserves due to wind power integration [14], [15]. A comprehensive review of strategies and studies on this topic can also found in the NREL report [16].

The easiest and fastest way to handle the integration of VREs that bring uncertainties and variabilities is to use the reserves. Using reserves does not require any additional tools than what the system operators already have. However, the use of reserves for VRE integration is inefficient and can only be used for a small percentage of VRE penetrations. Also, the energy commodity is different from the reserves in terms of price, trading, and hedging mechanisms that are in practice in the established markets today. Therefore, the reserves' use to fill in for the energy gap created by the VREs would create problems in the markets.

2.2. Chance-Constrained Programming

In optimization under uncertainty, one makes a decision using unknown parameters. It is possible that the realized scenario was not even considered when the decision was made and, therefore, could result in an unexpected situation. In most problems, there is a recourse one can follow to mitigate this unexpected situation. However, if one optimizes a problem where there are no recourses available, the decision-maker can guarantee feasibility as much as possible. The decision made guarantees the realization of an unexpected situation at a very low percentage. This

is to say that one or more constraints are enforced most of the time and are allowed to violate a diminutive percent of the time. Any constraint that contains a random variable or even a function of a random variable becomes a probabilistic constraint. The probability of a level at which the constraint enforced becomes a parameter. This level is known as reliability level; hence the Reliability Constrained Programming and Probabilistic Constrained Programming name used for Chance-Constrained Programming (CCP). The CCP uses the cumulative distribution function to transform probabilistic constraint into the deterministic equivalent of the optimization problem.

The earliest paper on CCP by Charnes and Cooper dates back to 1958 [17]. The application of CCP in UC was studied in [18]-[20]. Ozturk has carried out a research of CCP approach to stochastic UC in [21].

Chance-Constrained Formulation

$$\text{minimize } \mathbf{c}^T \mathbf{x} \tag{2.1}$$

$$\text{s. t. } \mathbb{P}[\mathbf{Ax} \geq \mathbf{b}] \geq \xi, \quad \xi \in [0,1] \tag{2.2}$$

Typical values for ξ would be 0.95, 0.99 etc.

2.3. Interval Optimization

The second approach is the application of Interval Optimization (IO), which uses confidence intervals in upper and lower bounds to represent the uncertainty. Unlike scenario-based optimization, which is discussed in the following section, IO does not hold any presumptions on probability distributions. The objective here is to achieve upper and lower bound feasibility rather than minimizing the cost.

One of the first applications of IO in UC was studied by Wang *et al.* in [22] for volatile node injections. In [23], Zhou *et al.* used IO in solving Stochastic-SCUC. Yu *et al.* applied IO to solve SCUC with high penetration of renewables [24]. In addition to using interval values for demand at each node, transmission contingencies are modeled as interval values with upper and

lower values. Essentially this reduces n constraints (for n - contingencies) to just one constraint. This guarantees that all n contingencies are feasible while significantly reducing the complexity of the problem. Also, UC with wind power integration using IO was studied in [25]. Comparison between IO and scenario-based optimization, which is discussed in the later part of this dissertation, was carried out by Wu *et al.* in [26].

Interval Optimization Formulation

$$\text{minimize } f(\mathbf{x}) := [f^L(\mathbf{x}), \quad f^U(\mathbf{x})] \quad (2.3)$$

$$\text{s. t. } \mathbf{x} \in C \quad (2.4)$$

where $C \subset \mathbb{R}^n$ is a nonempty set

$$f^L, f^U: \mathbb{R}^n \rightarrow \mathbb{R}$$

$$f^L(\mathbf{x}) \leq f^U(\mathbf{x}), \forall \mathbf{x} \in C$$

2.4. Robust Optimization

Robust Optimization (RO) is one of the two approaches to deal with uncertain data used in optimization (the other being stochastic optimization). Though roots of RO go to the 1970s, it has gained attention in the early 2000s. Ben-Tal and Nemirovski have authored several papers on this [27]-[29]. Gorissen *et al.* have published a practical guide to RO [30]. Theory and applications of Robust Optimization are given in [31] by Bertsimas *et al.* Unlike stochastic optimization, RO does not assume that probability distributions of data are known. For data, RO depends on *the uncertainty set*. RO assumes hard constraints for any realization of data in the *uncertainty set*. This *uncertainty set* is an essential part of RO. The method has gained popularity as it is computationally tractable and robust against all possible realizations of the modeled uncertainty. Application of RO in UC is given in [32]-[35]. However, the drawback is that RO optimizes for the worst-case scenario which is not realized often.

Adaptive Robust Optimization (ARO) remedies this drawback to an extent. Bertsimas *et al.* [36] and Ning and You [37] have applied ARO in solving UC problems. Other variants of RO are discussed in [38]. Zhao and Guan have applied a combined RO with stochastic optimization in solving the UC problem [39].

Robust Formulation

$$\text{minimize } \mathbf{c}^T \mathbf{x} \tag{2.5}$$

$$\text{s. t. } \mathbf{A}_i \mathbf{x} \geq \mathbf{b}_i, \quad \forall \mathbf{A}_i \in U_{A_i}, \quad \forall \mathbf{b}_i \in U_{b_i}, i = 1, \dots, m \tag{2.6}$$

Where U_a and U_b are given uncertainty sets

2.5. Stochastic Optimization

Finally, a common approach presented in the literature is the use of stochastic optimization (SO), namely stochastic NCUC (S-NCUC), in which the operating cost of the expected system is minimized across several scenarios, each representing a possible realization of uncertainties. S-NCUC is typically a large-scale, non-convex, mixed-integer problem. The problem is formulated in two or more stages, and either stage-wise or scenario-wise decomposition technique is used in solving the problem. Benders decomposition is a stage-wise (cut-based) method, while Progressive Hedging Algorithm (PHA) and Dual decomposition (Lagrangian relaxation based) are scenario-wise examples.

2.5.1. Stage-Wise Decomposition

The S-NCUC is often modeled as a two-stage stochastic problem where the first-stage unit commitment decisions (here and now decision) are the same for all scenarios. The second stage depends on the realization of any scenario. J. F. Benders proposed partitioning procedures for solving mixed variable programming problems in 1961 [40]. Van Slyke and Wets applied Benders Decomposition (BD) to stochastic programming in 1969 [41]. BD involves creating a master problem (MP) and sub-problem (SP). The MP makes first-stage decisions (unit commitment) and

passes them on to SP. With fixed first stage decisions, SP solves scenarios, generates new cuts (constraints), and passes the further cuts to MP. This interaction between MP and SP continues until convergence.

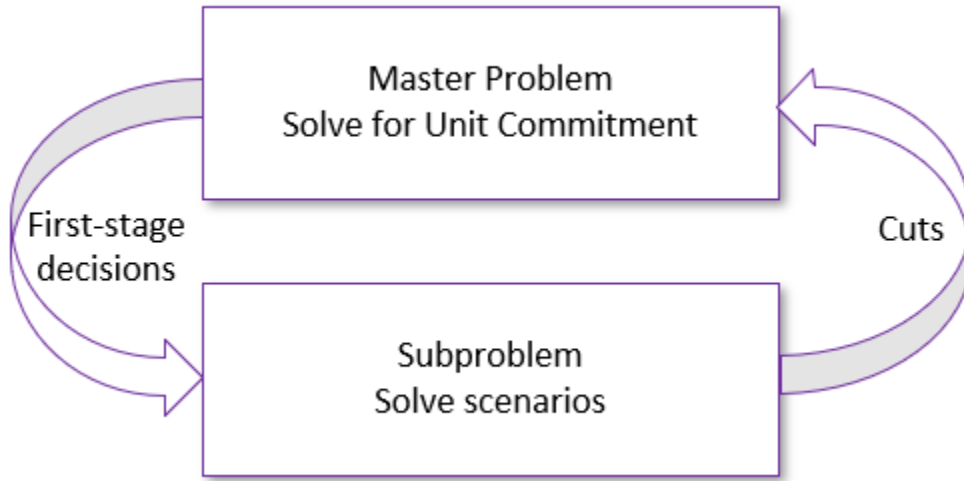


Figure 2.1 Bender Decomposition showing interaction between Master and Subproblems.

Baptistella and Geromel proposed one of the early applications of BD to solve UC [42]. A tutorial on BD in restructured power systems [43] is the right place for beginners. One advantage of BD is that both the lower and the upper bound are obtained as part of the solution. The disadvantage is the difficulty involved in solving large problems. As the iteration progresses, the MP gets increasingly difficult to solve, especially for a massive problem like the one used in the study cases in this research. The other issue with BD is parallelization. Though BD lends itself to parallel computation, the MP growth makes parallelization an unbalanced one. There have been several enhancements in the literature on improving BD-based methods when applied to S-NCUC problems [44], [45]. To solve multi-stage problems, one can use nested BD [46], [47].

2.5.2. Scenario-wise decomposition

In this decomposition method, the scenarios are solved individually by applying Lagrangian to decouple the linkage between scenarios. Carøe and Schultz have applied the dual decomposition method to solve stochastic integer programming in [48]. Takriti *et al.* have used Lagrangian decomposition to solve the Stochastic UC problem [49]. If one has to decide at each

stage, often called here-and-now decisions, it must consider all the possible scenarios that are considered. In the PSOP problem, this is a unit commitment decision. The other name for such a decision is “*Implementable Solution.*” Wu *et al.* applied the same Lagrangian method used to solve S-NCUC in [50].

The penalty function added to the Lagrangian method (augmented Lagrangian) is used in Progressive Hedging Algorithm (PHA). Initially proposed by Rockafellar and Wets [51], PHA to solve stochastic problems involving continuous variables has been applied to solve a problem involving integers along with continuous variables. Løkketangen and Woodruff first used the PHA to solve mixed-integer multistage stochastic programs [52]. Fan and Liu applied PHA to solve the stochastic transportation network problem [53]. Nevertheless, PHA is initially devised for the continuous convex program and is not provably convergent for non-convex problems such as UC. Watson and Woodruff made heuristic-based novelties to mitigate or solve issues related to PHA application to MIP [54]. This PHA will be discussed in the Chapter 4 of this report.

Like stage-wise decomposition, scenario-wise decomposition methods can also be solved in parallel. The advantage of the scenario-wise decomposition method compared to the stage-wise decomposition method is the uniform distribution of sub-problem difficulty. The computational difficulty of MP in the stage-wise method can grow significantly as the iteration progresses. It has been observed that the time taken to solve each scenario problem takes less time as the iteration progresses in scenario-wise decomposition. Also, scenario-wise problems can be implemented as a wrapper over an existing NCUC algorithm in use. This wrapper implementation is an attractive one as it does not take much time and cost to implement, especially if one desires to have proof-of-concept. However, the major drawback of scenario-wise decomposition is that obtaining a lower bound to measure the solution quality requires solving a particular problem. To get a lower bound for PHA, Gade *et al.* [55] have proposed a method that uses dual prices from PHA. Lower bound can be obtained in any iteration of PHA by solving a different problem simultaneously. However, such a lower bound obtained is sensitive to the penalty factor chosen. This drawback is mitigated by Boland *et al.* by combining PHA with the Frank-Wolfe method. This method is not sensitive to the penalty factor [56]. An application of the Frank-Wolfe method to obtain a lower bound for the S-NCUC problem is carried out in [57].

Two-stage Stochastic Programming Formulation

$$\text{minimize } \mathbf{c}^T \mathbf{x} + \mathbb{E}[\mathbf{q}^T \mathbf{y}] \quad (2.7)$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{b}, \quad \mathbf{x} \geq 0 \quad (2.8)$$

$$\mathbf{Tx} + \mathbf{Wy} = \mathbf{h}, \quad \mathbf{y} \geq 0 \quad (2.9)$$

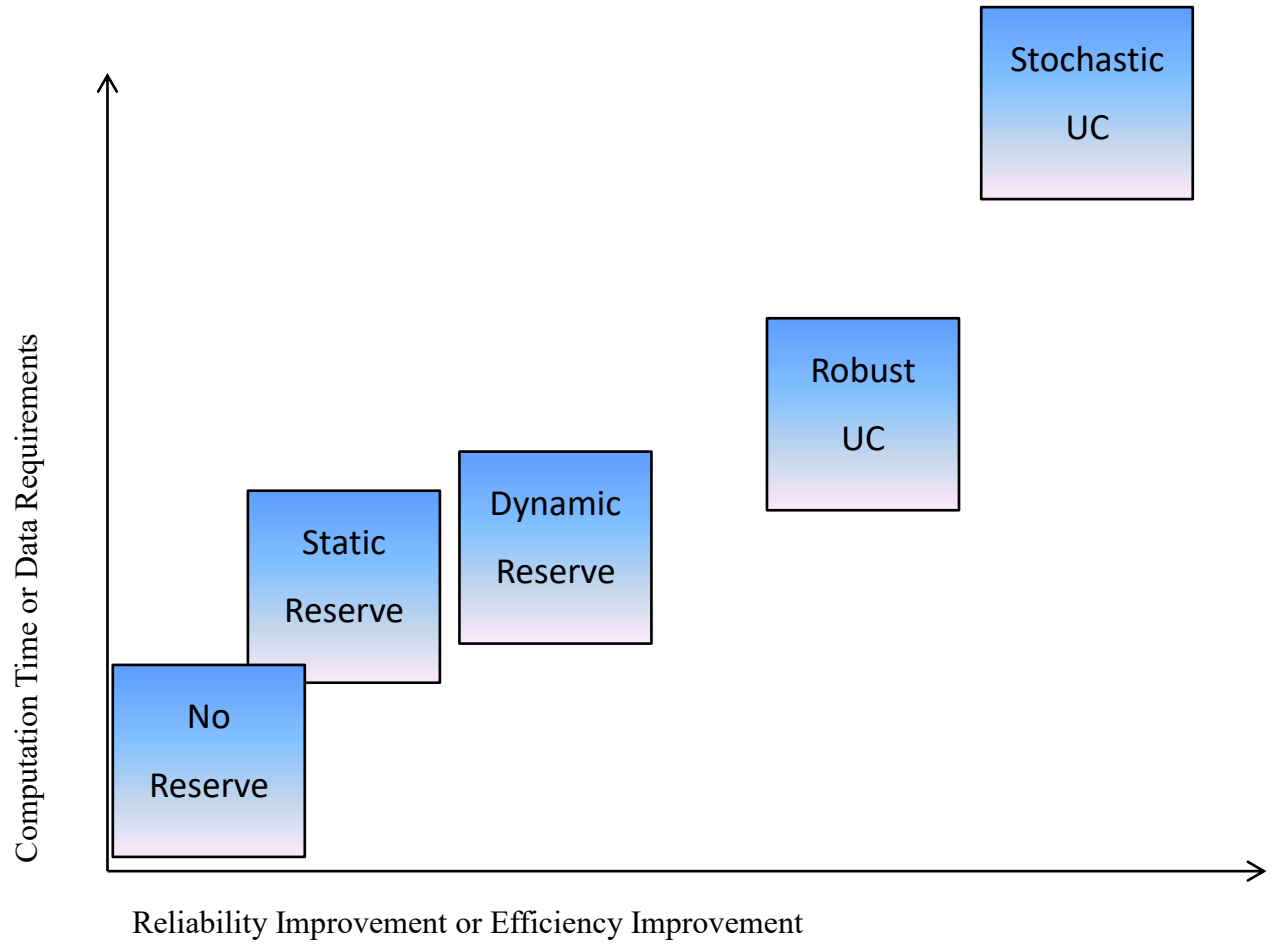


Figure 2.2 Reliability vs computation efforts for different methods

2.6. Hybrid Solutions

Figure 2.2 depicts the reliability improvement versus the computational time of the different methods considered so far. Researchers have combined two or more different methods discussed earlier and applied them in solving S-UC problems. Such approaches bring the best of those methods. Colonetti and Finardi combined Lagrangian relaxation and Benders decomposition in [58] to solve the stochastic hydrothermal UC problem. In [59], Zhao and Guan applied unified stochastic and robust optimization techniques in solving the UC problem.

2.7. Scope of The Research

Motivated by the challenge posed by the VRE as explained in Chapter 1 and given different ways of taking up this change as described in this chapter, the scope of this research is defined with the following three conditions to make the research applicable for the real-world situations:

1. The solution should not be an exogenous one.
2. The solution should be economically optimal and still keeps the reliability of the power system intact.
3. The solution should be easy to implement with the least cost, time and resources.

Applying these conditions to the different methods discussed earlier, condition 1 eliminates dynamic reserve-based approaches discussed in 2.1 while condition 2 eliminates all other methods discussed in 2.2 through 2.4. The last condition eliminates stage-wise decomposition methods such as BD. This leaves us with scenario-wise decomposition methods.

3. Stochastic-Network Constrained Unit Commitment

This chapter starts with a brief introduction to Unit Commitment (UC), followed by formulating the Stochastic-Network Constrained Unit Commitment (S-NCUC) problem. Section 3.3 is dedicated to the computational environment used in the simulation, followed by Section 3.4 in which three different systems used in the simulation are discussed. Scenario generation is a crucial part of any stochastic programming; therefore, Section 3.5 is dedicated to this topic. Finally, Extensive Formulation results are presented.

3.1. Unit Commitment (UC)

It is well known that the electricity demand must be met with supply in real-time. Since the thermal generators have a long lead-time between the time the decision is made to bring the unit online and synchronizing the unit to the grid, the operational planning occurs well in advance of real-time operations. During this operational planning, a generation commitment decision must be made to meet the forecasted demand by committing enough generators. This commitment decision is an economical one. The total cost of serving the demand is minimized while subject to individual unit constraints and system-wide power balance, and other constraints. Near the real-time operation, another decision is made to commit, or de-commit quick-start generation units (QSG) called real-time commitment (RTC). During the real-time operation, the final decision is made by dispatching the committed units to meet the actual demand and is called economic dispatch (ED). Like UC, both RTC and ED are based on an objective where a low-cost solution is sought after. Wood and Wollenberg's text [60] gives an excellent introduction to UC and ED. Unit Commitment problem is a scheduling problem.

Network Constrained UC (NCUC) and Network Constrained Economic Dispatch (NCED) are UC and ED with additional constraints stemming from the thermal and dynamic limits of transmission lines. It is crucial to control the transmission grid's flows that do not push the actual flows beyond thermal or dynamic limits, thereby causing system security problems. Nowadays such violations are avoided not only on the base case but also under $n-k$ contingencies. When such contingencies are considered, the UC problem is called Security Constrained UC (SCUC). Though the stochastic problem formulated and solved is for NCUC, the formulation can also be applied to SCUC without any changes.

3.2. S-NCUC Formulation

Since a plethora of papers, books and dissertations are available on the deterministic version of UC [61]-[66] and the focus is on stochastic formulation, the discussion starts straightaway into the stochastic version. An S-NCUC formulation that originated from the Flexible Energy Scheduling Tool for Integrating Variable Generation (FESTIV) [67], [68] is used in the research. The objective of the two-stage S-NCUC is to minimize the expected system operating cost across the different scenarios as follows:

Objective function:

$$\begin{aligned} \zeta = \underset{I, IU, p}{\text{minimize}} \quad & \sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left(NL_i \cdot I_{i,t} + SU_{i,t} \cdot IU_{i,t} + \sum_{s=1}^S Pr^s \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) \right) \right. \\ & + \sum_{s=1}^S Pr^s ((AL_t^s + LL_t^s) \cdot VOLL) \\ & \left. + \sum_{s=1}^S Pr^s \left(\sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right) \right] \end{aligned} \quad (3.1)$$

where

$$I \in \{0,1\}^{NI \times NT \times S}$$

$$p \in \mathbb{R}^{NG \times NI \times NT \times S}$$

The objective function in (3.1) has commitments ($I_{i,t}$), startups ($IU_{i,t}$) and dispatch ($p_{d,i,t}^s$) decisions that use corresponding no-load cost (NL_i), startup cost ($SU_{i,t}$) and dispatch cost based on offer curve ($IC_{d,i}$). The startup decision is independent of scenarios as it is a here and now decision; it is summed up over generators and intervals. However, the dispatch decision depends on the scenarios considered. Therefore, it is summed up over all the generators, intervals, and scenarios.

The second part of the objective function (3.1) includes two types of penalty. One (VOLL) arises from the violation of the power balance constraint (loss-of-load (LL_t^s) or additional-load slack (AL_t^s)). The other (VOOB) is transmission violation constraint (branch slack in either direction ($BrSl1_{l,t}^s, BrSl2_{l,t}^s$)). These are used to obtain a solution; otherwise they would result in infeasibility.

It should be noted that the probability of realization of each scenario is multiplied by the cost of that scenario. The probability of all the scenarios should add up to one.

Subject to the following constraints:

Power balance constraint

$$\sum_{i=1}^{NI} Pg_{i,t}^s = \left[\sum_{n=1}^N D_{n,t}^s - LL_t^s + AL_t^s \right] \quad \forall t \in NT, \forall s \in S \quad (3.2)$$

This power balance constraint is a system-wide constraint, meaning any sum of all the generation in the system must be equal to the sum of the system's demand. This constraint forces the supply to meet the demand for each interval. As discussed earlier, loss-of-load and additional-load slack variables are there to obtain a feasible solution by applying a penalty.

Power flow constraints

$$\left. \begin{aligned} LF_{l,t}^s &\leq \overline{LF}_l + BrSl1_{l,t}^s \\ LF_{l,t}^s &\geq -\underline{LF}_l + BrSl2_{l,t}^s \end{aligned} \right\} \quad \forall l \in L, \forall t \in NT, \forall s \in S \quad (3.3)$$

The transmission lines must not be loaded above its thermal or stability limit in either direction. Such an overload can cause permanent sag of the line violating the sag limit. This constraint enforces such restriction for each transmission line in the system.

Power flow equation

$$LF_{l,t}^s = \sum_{n=1}^N SF_{l,n} \cdot \left(\sum_{i=1}^{NI} Pg_{i,t,n}^s - D_{n,t}^s + LL_t^s - AL_t^s \right) \quad \forall l \in L, \forall t \in NT, \forall s \in S \quad (3.4)$$

Generator minimum and maximum output constraints

$$I_{i,t}^s \cdot \underline{Pg}_i \leq Pg_{i,t}^s \leq \overline{Pg}_i \cdot I_{i,t}^s \quad \forall i \in NI, \forall t \in NT, \forall s \in S \quad (3.5)$$

Constraint (3.5) enforces that the generators are not dispatched below the low limit or above the high limit. Typically, large fossil fuel generators have a low limit needed to make sure a minimum amount of generation to supply auxiliary load such as boiler water feed pump, draft fan, etc.

Generator ramp up and ramp down constraints

$$\left. \begin{aligned} Pg_{i,t}^s - Pg_{i,t-1}^s &\leq RU_i \\ Pg_{i,t-1}^s - Pg_{i,t}^s &\leq RD_i \end{aligned} \right\} \quad \forall i \in NI, \forall t \in NT, \forall s \in S \quad (3.6)$$

Ramp up (RU_i) and ramp down (RD_i) constraints ensure that the generator can meet the demand from one interval to the next by increasing/decreasing a certain amount of output in a specific time. These ramp-up/down rates could be different from startup and shutdown rates.

Generator minimum up time and minimum down time constraints

$$\left. \begin{aligned} \sum_{t=T}^{T+TU_i-1} I_{i,t}^s &\geq TU_i \\ \sum_{t=T}^{T+TD_i-1} (1 - I_{i,t}^s) &\geq TD_i \end{aligned} \right\} \quad \forall i \in NI, \forall t \in NT, \forall s \in S \quad (3.7)$$

A typical fossil-fuel generator has a constraint where it will have to be online for a minimum number of hours once it comes online (TU_i). When offline, the generator will have to remain offline for a minimum number of hours before it comes online again (TD_i). Enforcement of these minimum up and downtime constraints is accomplished in constraint (3.7).

In the S-NCUC formulation, shift factors are used to calculate the line power flow. The shift factor and power flow calculations can be found in [68]. Other prevailing constraints such as

spinning and non-spinning reserve constraints, segment generation limits are also considered. In addition, contingency constraints can be taken into consideration in the S-NCUC formulation. The feasible region of decision variables is given as:

$$\{I_{i,t}, p_{d,i,t}^s\} \in \Lambda^s, \forall i \in NI, \forall t \in NT, \forall d \in NG \quad (3.8)$$

where Λ^s is determined by all the above constraints. Problem (3.1), constraints (3.2) - (3.7) are also known as the Extensive Form (EF) of the two-stage S-NCUC, in which even a moderate number of scenarios can result in a computational burden that quickly exceeds the capability of any state-of-the-art MIP solver. Besides, the computational burden increases exponentially with the size of the problem using the branch-and-cut method [69]. This increased computational burden is why scenario-based decomposition frameworks such as Lagrangian relaxation are used to solve a large-scale S-NCUC problem iteratively [70].

3.3. Computational Environment

The proposed algorithms are implemented in MATLAB within the National Renewable Energy Laboratory (NREL)'s Flexible Energy Scheduling Tool for Integrating Variable generation (FESTIV) framework. The S-SCUC is modeled in General Algebraic Modeling System (GAMS) [71] and all the problems were solved using CPLEX [72]. Figure 3.1 shows data flow between MATLAB where iterations takes place and GAMS which does optimization. Figure 3.2 depicts the FSTIV graphical user interface. The computing platform has 256 GB RAM, Intel Xeon CPU E5-2640 with dual processors.

FESTIV Framework

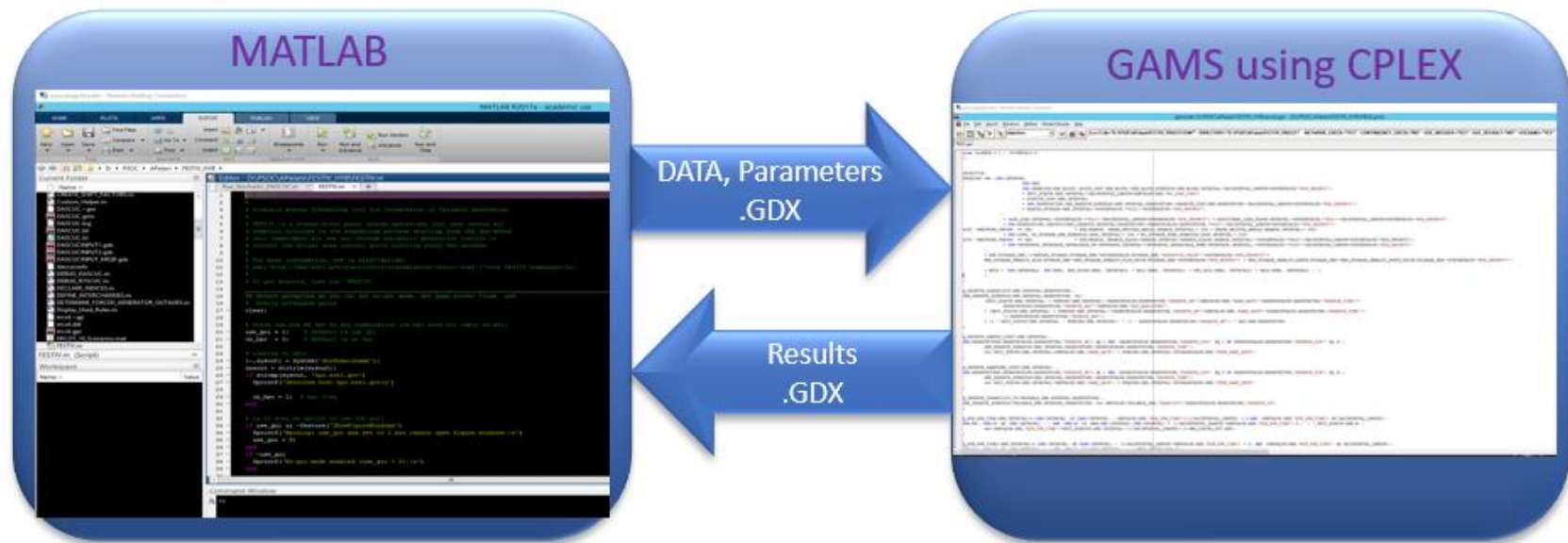


Figure 3.1 NREL's FESTIV Framework

FESTIV

Flexible Energy Scheduling Tool for Integrating Variable generation

FESTIV

Input File:

IIT_118bus_CAISO_basecase_err00_Jan.h5

Browse

☐ auto save

DASCUC

t_DAC: 24 I_DAC: 1

H_DAC: 24 P_DAC: 1

G_DAC: 12 H_Type: Type 1

Load Forecast

2 - Perfect Forecast

VG Forecast

1 - From Data File

Reserve Levels

1 - No reserve

DASCUC Rules

RTSCUC

t_RTC: 15 I_RTC: 15

H_RTC: 12 P_RTC: 15

t_RTCSTART 1

Load Forecast

2 - Perfect Forecast

VG Forecast

1 - From Data File

Reserve Levels

1 - From Day Ahead

RTSCUC Rules

RTSCED

t_RTD: 5 I_RTD: 5

H_RTD: 5 P_RTD: 5

I_RTD_ADV: 15

Load Forecast

2 - Perfect Forecast

VG Forecast

1 - From Data File

Reserve Levels

1 - From Day Ahead

RTSCED Rules

AGC

AGC Param

AGC Rules

Simulation Time (D:H:M:S)

1 : 0 : 0 : 0

Network Check

☒ Yes ☐ No

CTGC Check

☐ Yes ☒ No

Extra Options

RPU

Multiple Runs

CTGC Options

Debug

Other Rules

Save Rules

Load Rules

GAMS

Cancel

Go!

Figure 3.2 FESTIV - user interface²

² [Flexible Energy Scheduling Tool for Integrating Variable Generation | Grid Modernization | NREL](#)

3.4. Systems Modeled/Used in Simulations

Three systems differing in size are used in the research. The system chosen varies in size and are discussed in detail in this section.

- RTS-96 System
- IEEE 118-bus System
- ERCOT-Like System

3.4.1. RTS-96 System

The IEEE Reliability Test System (RTS-96) was developed [73], [74] by modifying and updating the original IEEE RTS (RTS-79). This system is widely used in experiments for all kinds of bulk power-system related studies. The current research uses the modified single-area that has 24 buses, 38 branches and 26 generators. The average load is around 1200 MW and the peak load is nearly 2500 MW. A wind turbine generator (WTG) is connected to Bus 23 with an installed capacity of 130 MW, which is about 4% of the total capacity in the system. One-line diagram of the RTS-96 System is provided in Figure 3.3.

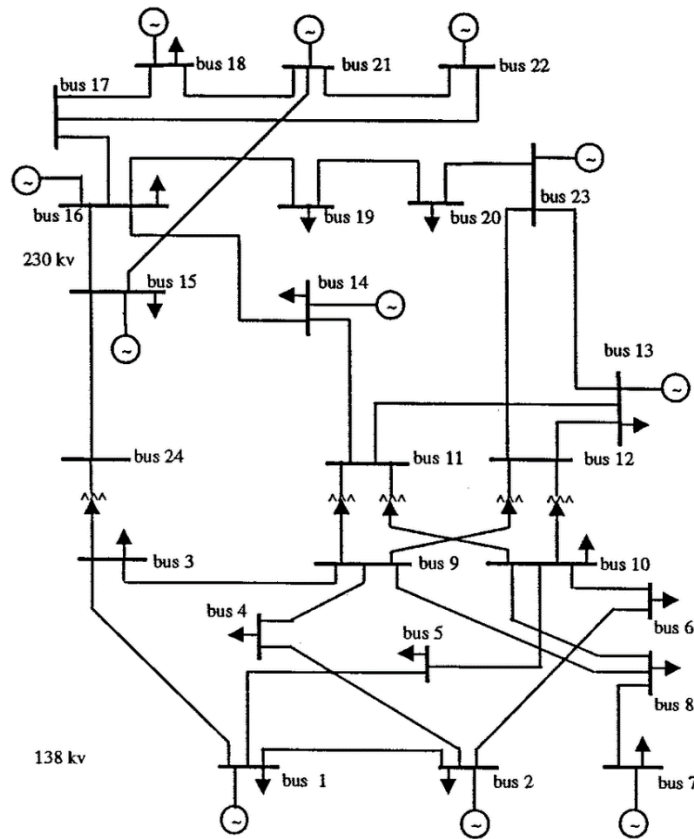


Figure 3.3 The IEEE Reliability Test System (RTS-96) diagram [74]

To get an understanding of the problem size of RTS-96 mathematical model is done by looking at the GAMS statistics in terms of number of equations, discrete variables, etc. Statistics for RTS-96 is shown in Figure 3.4. There are 2,232 first stage decisions over a 24-hour period for this system.

```
GAMS 28.1.0 r5b48834 Released Aug 2, 2019 WEX-WEI x86 64bit/MS Windows
General Algebraic Modeling System
Model Statistics SOLVE SCUC Using MIP From line 1695

MODEL STATISTICS

BLOCKS OF EQUATIONS      108      SINGLE EQUATIONS      191,065
BLOCKS OF VARIABLES      25       SINGLE VARIABLES      160,729
NON ZERO ELEMENTS        804,796  DISCRETE VARIABLES     2,232
```

Figure 3.4 GAMS model statistics for the RTS-96 System

The research uses the RTS-96 and IEEE 118-bus system to establish unique features of FW-PHA and PBGS through results and arrive at a conclusion that will be used to interpret the large system result as big as Electric Reliability Council of Texas (ERCOT).

3.4.2. The IEEE 118-bus System

A modified IEEE 118-bus System [75], [76] has 118 buses, 186 branches and 54 generators that includes 10 WTGs. These WTGs are located at buses 4, 26, 27, 40, 49, 62, 89, 100, 107 and 112. The installed capacity is 376 MW (8% of the total capacity). The network diagram of this system is shown in Figure 3.5. GAMS statistics is given in Figure 3.6.

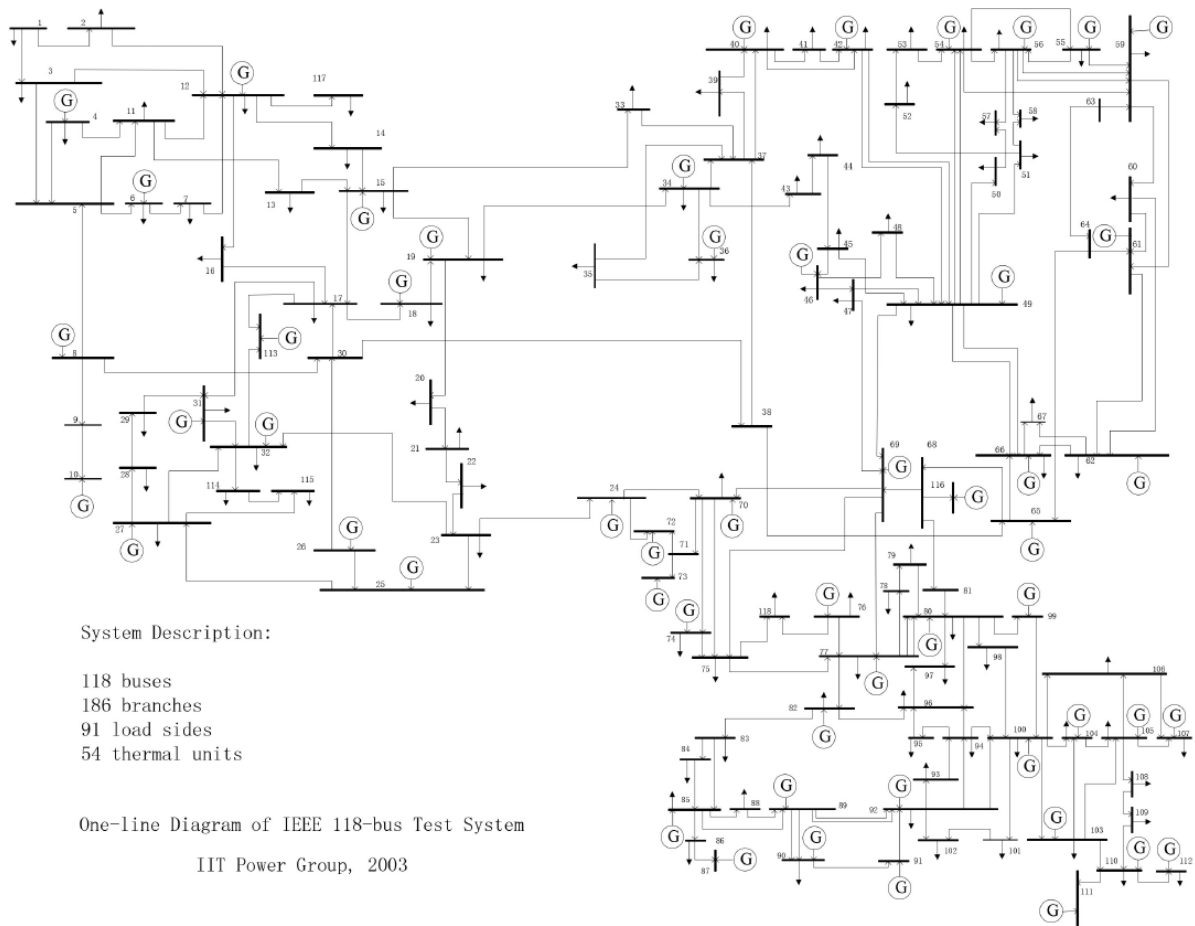


Figure 3.5 The IEEE 118-bus one-line diagram³

³ [Visio-IEEE_118bus_54T.vsd \(iit.edu\)](#)

```

GAMS 28.1.0 r5b48834 Released Aug 2, 2019 WEX-WEI x86 64bit/MS Windows
General Algebraic Modeling System
Model Statistics      SOLVE SCUC Using MIP From line 1695

MODEL STATISTICS

BLOCKS OF EQUATIONS      108      SINGLE EQUATIONS      463,344
BLOCKS OF VARIABLES      26      SINGLE VARIABLES      448,369
NON ZERO ELEMENTS      5,410,126      DISCRETE VARIABLES      4,032

```

Figure 3.6 GAMS model statistics for the IEEE 118-bus System

3.4.3. The ERCOT-like Large System

This system is a redacted version of the ERCOT system with 7226 buses, 8853 branches and 725 generators that includes 178 wind farms. The names of buses, loads, generators along with generation costs are redacted in the system modeled. In fact, it is so much redacted and modified that it resembles the ERCOT system from the network and size point-of-view only. The installed capacity is over 80,000 MW of which 21,582 MW is from wind. An overview of the transmission network of the system is shown in Figure 3.7. The transmission system has three different levels of voltages. At the highest level is 345 KV (red) followed by 138 KV (blue) and 69 KV (green).

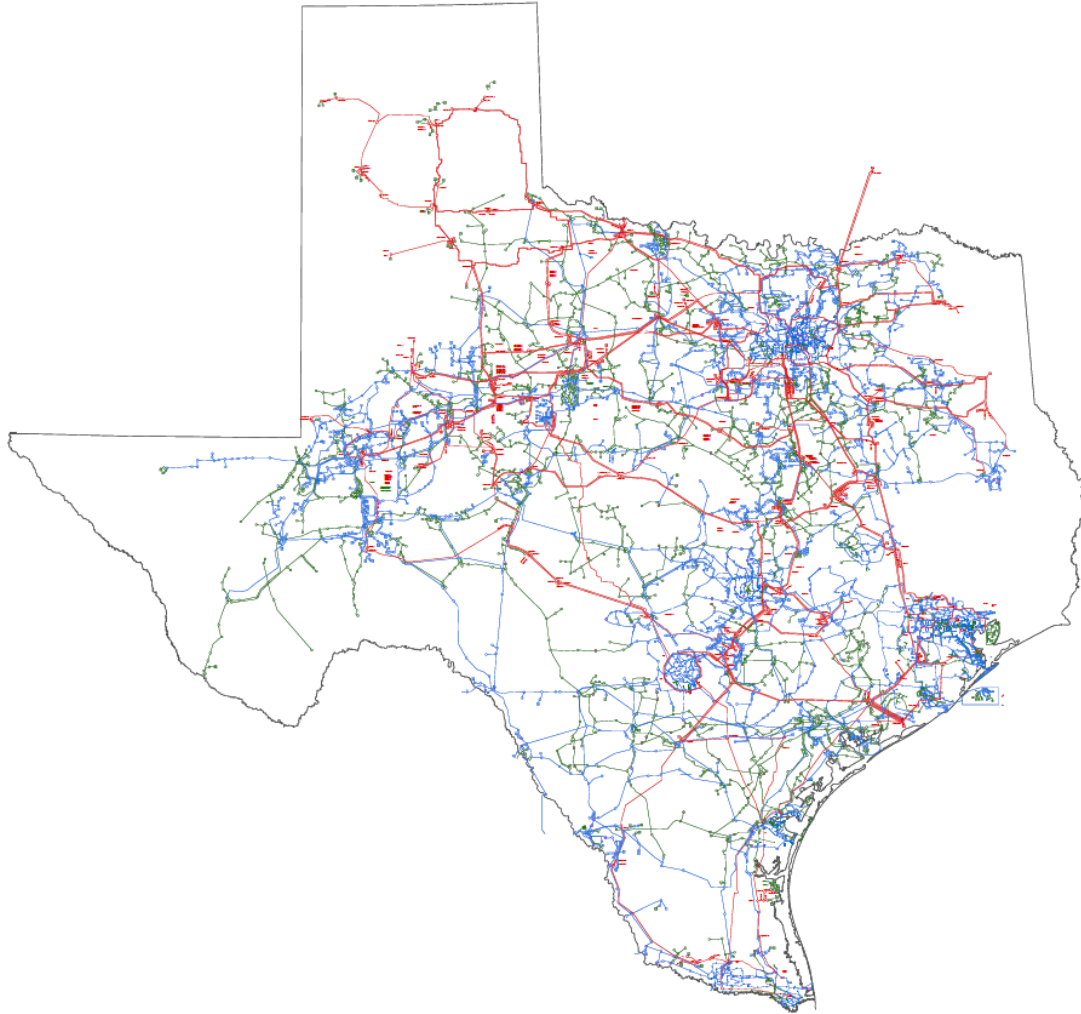


Figure 3.7 Bird's-eye View of the transmission system of ERCOT⁴

This large system is too big to be solved even deterministically as the number of variables and number of equations are close to 2 million each, and the total non-zero elements are over a billion. Figure 3.8 shows the model statistics.

⁴ <https://mis.ercot.com>

```
GAMS 24.7.4 r58773 Released Sep 19, 2016 WEX-WEI x86 64bit/MS Windows
General Algebraic Modeling System
Model Statistics SOLVE SCUC Using MIP From line 11952
```

MODEL STATISTICS

BLOCKS OF EQUATIONS	122	SINGLE EQUATIONS	1,654,327
BLOCKS OF VARIABLES	25	SINGLE VARIABLES	1,835,137
NON ZERO ELEMENTS	1,199,236,700		

Figure 3.8 GAMS model statistics for the ERCOT-Like System

In order to solve such a large system, it is necessary to eliminate the constraints that will not be active for a given network topology, load and generation. Over 95% of the 8853 branches are inactive for the scenarios generated and studied in this research. For each scenario, the active constraints were identified by first solving the NCUC problem without network constraints, and from the result a list of branches that violated its limit as active constraints is compiled. The NCUC problem is solved again with network constraints using these active constraints and screened for additional branch violations. This process was repeated until violations were eliminated or no new violations were found. Flow chart of this process is depicted in Figure 3.9.

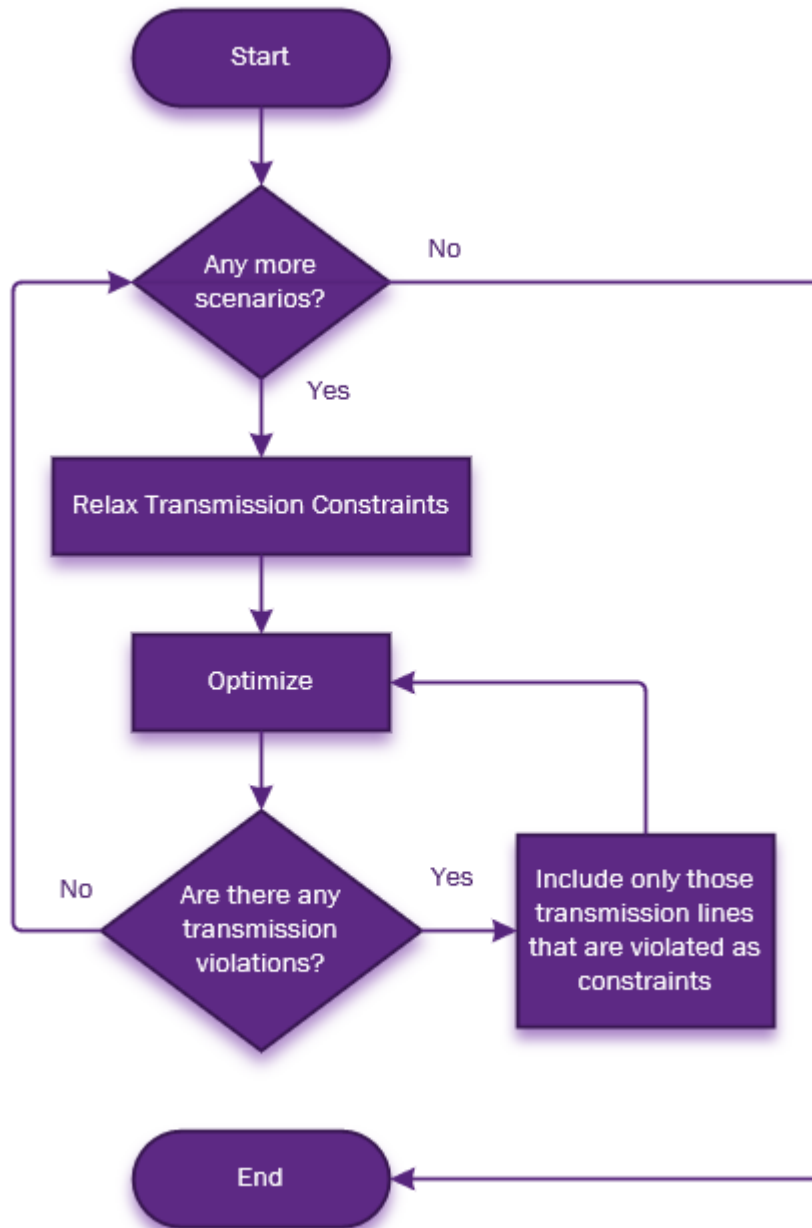


Figure 3.9 Line screening for the transmission constraints

An efficient method of identifying inactive constraints is given in [77]. Once all the active transmission constraints are identified for each scenario, combined active constraints would be used as the only transmission constraints in further simulations.

3.5. Scenario Generation

The modeling of uncertainty is crucial in stochastic programming. There are many techniques available, and depending on the technique used, the modeling of the uncertainties could be different. The goal of scenario generation is modeling of the uncertainties that represent a possible outcome. In a two-stage problem, the scenarios are for the second stage where any of those scenarios could be realized. In a multi-stage problem, a scenario tree is needed and each stage could have its own set of scenarios. For the S-SCUC problem, scenario generation should be related to the forecasted value of variables. This is particularly important for wind generation forecast as it is based on the wind speed forecast. Wind speed is based on the location and time, which must be included in the scenario generation. The impact of location and time on the generation output is not only applicable for wind generation but also for solar generation. Scenarios must be generated for each wind/solar farm. This is especially true for a system that geographically spans hundreds of miles.

The performance of S-SCUC is driven by how well the selected scenarios represent the stochastic nature. Too many scenarios would be time consuming to solve, and too few may not represent the uncertainties well. Naturally, one can think that a large number of scenarios would yield higher quality of solution. This may be true up to a point after which the quality of solution might not improve for the increased number of scenarios. This is where scenario reduction techniques come into help. Scenario reduction technique [78] can be applied to bundle two or more similar scenarios based on certain probabilistic metrics to reduce the number of scenarios to be considered in the problem. Comparison of different techniques are presented in [79]. Several papers are devoted to the scenario generation topic [80]-[83] and research is actively conducted in this area.

Since the research focus is not about scenario generation or reduction, there will not be any discussion on the methods used other than the one employed in generating scenarios for the simulations in this research. The scenario generation technique with autoregressive moving average, i.e., ARMA (1, 1), is used to generate scenarios [84], each representing the possible realization of load and wind condition for all the three systems considered. The ARMA (1, 1) used is

$$e_t = \alpha \cdot e_{t-1} + \beta \cdot L_{t-1} + L_t \quad (3.9)$$

where e is the forecast error of wind generation or load at time t , and L_t is a normal-distribution function with a varying standard deviation of the load forecast and the wind generation forecasts. α and β are ARMA parameters determining the degree to which the previous value influences the current value. One set of 10 scenarios for RTS-96 was generated. For the IEEE 118-bus System, a total of 27 sets were generated. One set of 10 scenarios and 26 sets with 50 scenarios. For the ERCOT-like System, one set of 30 scenarios were generated. This limitation of 30 scenarios was due to the computational constraints. However, larger deviations in both load and wind generations were used in scenario creation. Table 3.1 shows each scenario set with the standard deviation (STD) used for load forecast and wind generations.

Table 3.1 List of scenarios sets with scenario generation statistics

Scenario Sets	Load Forecast Deviation		Wind Forecast Deviation		Number of Scenarios
	STD	% Dev	STD	%Dev	
RTS96-10-S1	3%	23.58	6%	45.16	10
IEEE118-10-S1	3%	23.58	6%	45.16	10
IEEE118-50-S0	3%	23.58	6%	45.16	50
IEEE118-50-S1	4%	31.08	8%	57.62	50
...
IEEE118-50-S25	4%	31.08	8%	57.62	50
ERCOT-30-S1	6%	45.16	10%	68.26	30

Figures 3.10 through 3.24 show the load, wind and selected individual wind generation scenarios for each of the three systems.

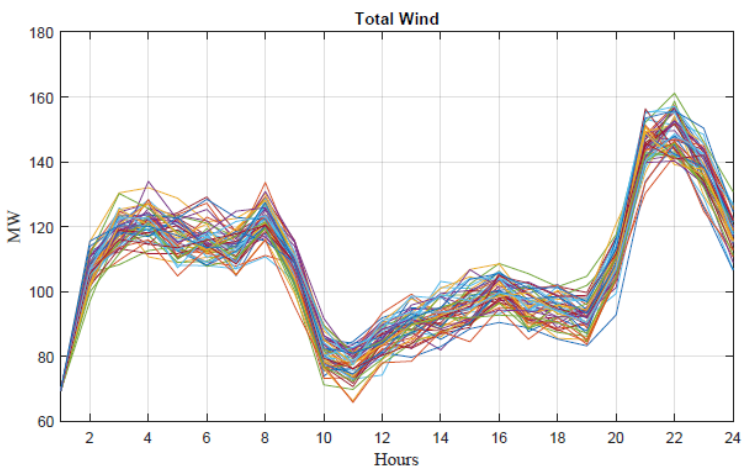
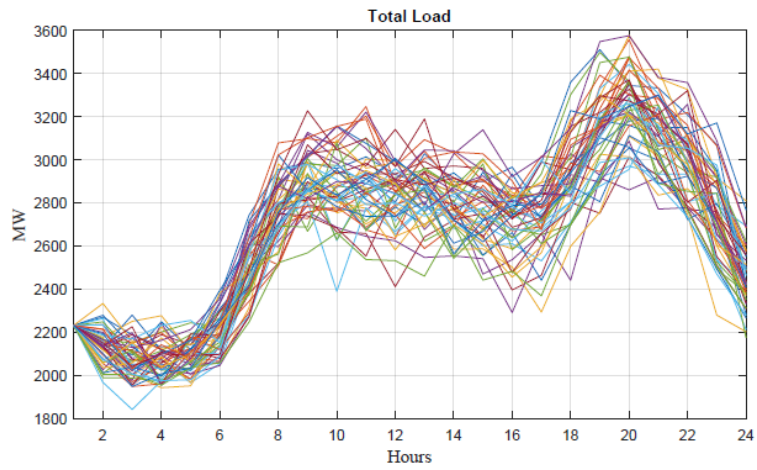


Figure 3.10 The IEEE 118-bus System set 1 - load and wind, 50 Scenarios

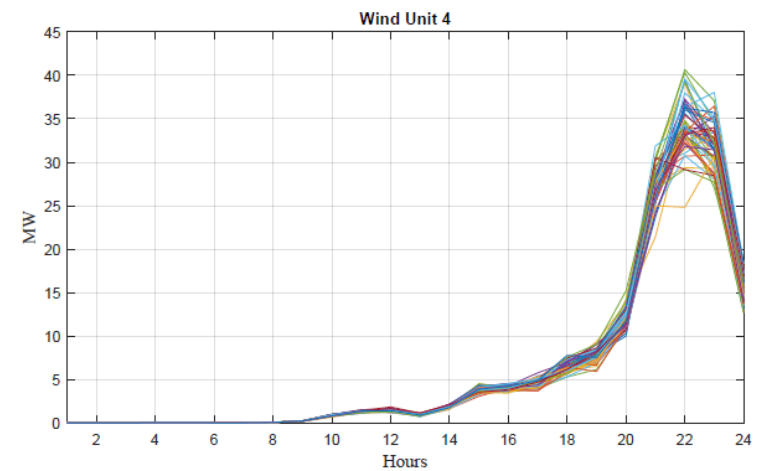
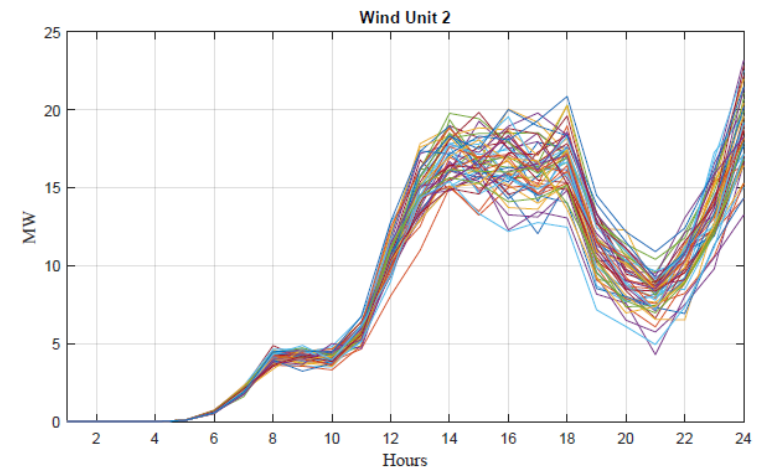


Figure 3.11 The IEEE 118-bus System set 1 – WTG 2 and 4, 50 scenarios

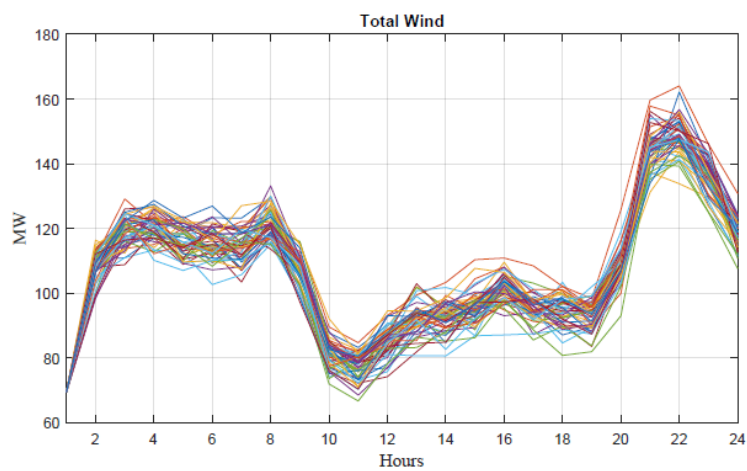
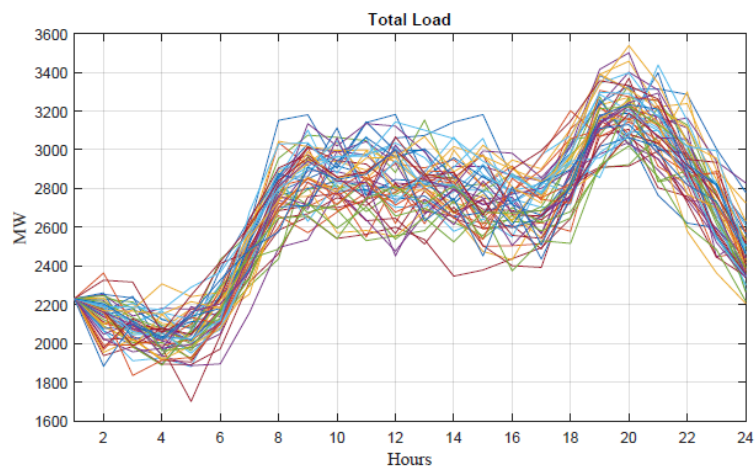


Figure 3.12 The IEEE 118-bus System set 6 - load and wind, 50 scenarios

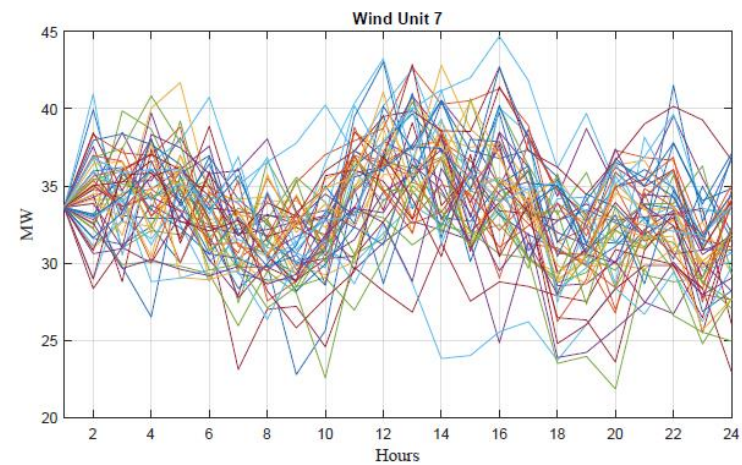
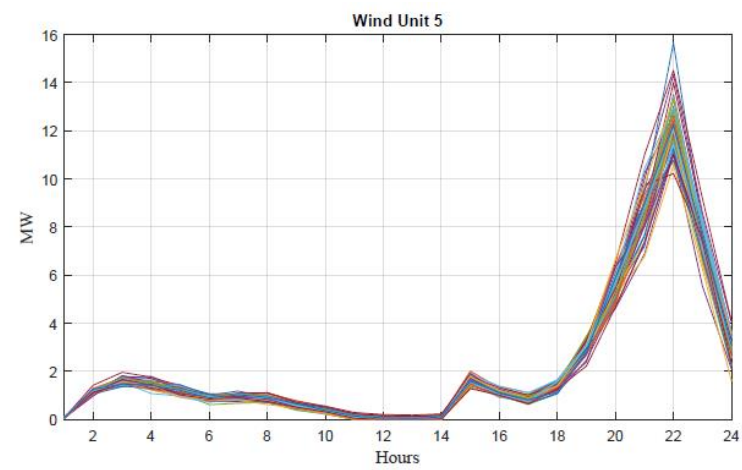


Figure 3.13 The IEEE 118-bus System set 25 – WTG 5 and 7, 50 scenarios

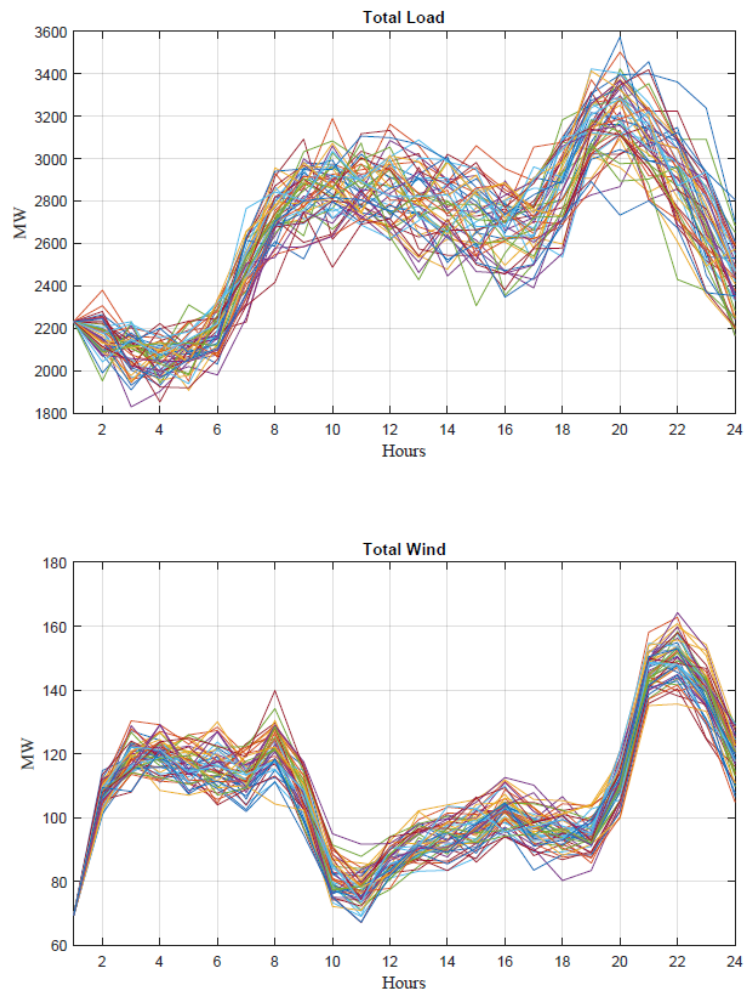


Figure 3.14 The IEEE 118-bus System set 14 - load and wind, 50 scenarios

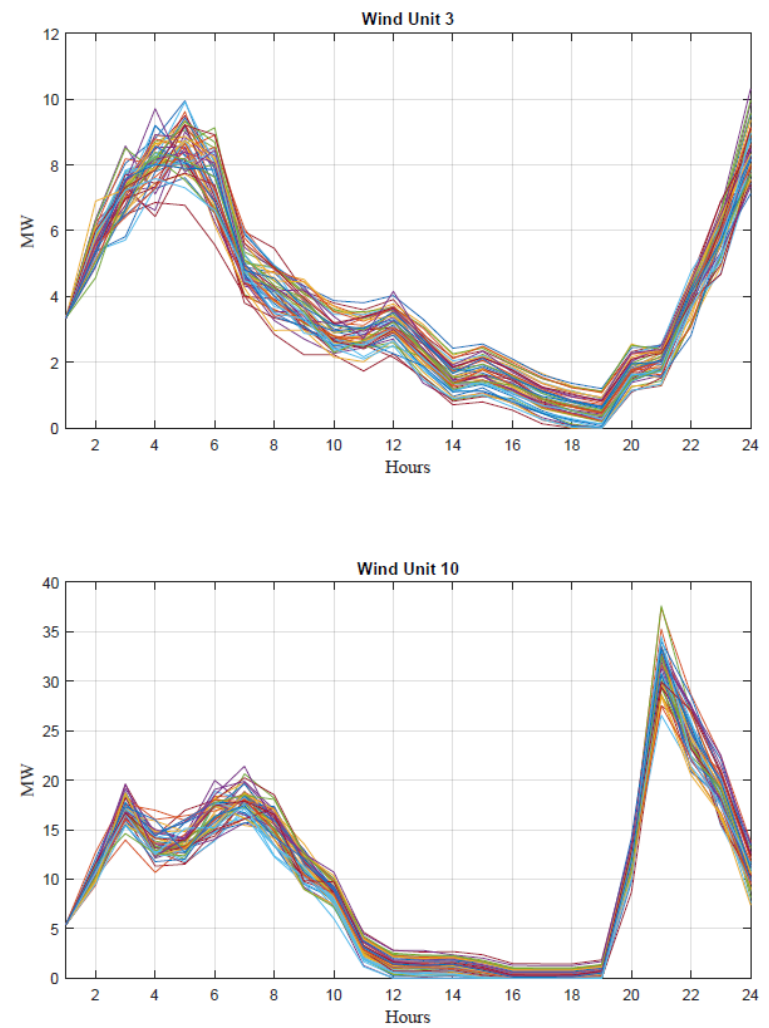


Figure 3.15 The IEEE 118-bus System set 14 – WTG 3 and 10, 50 scenarios

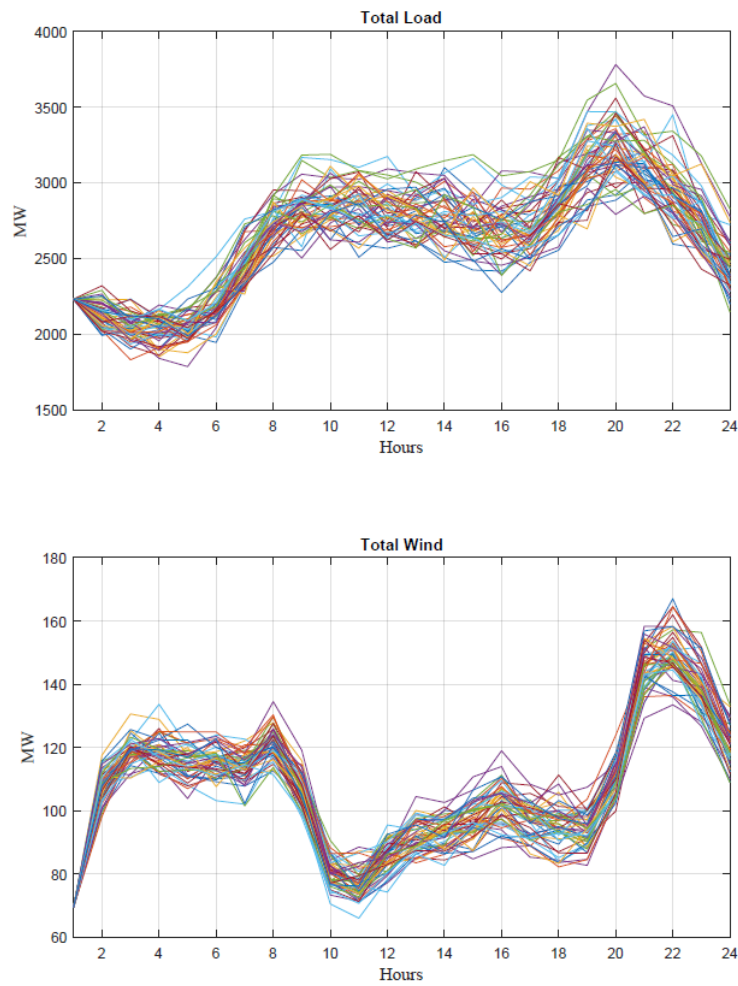


Figure 3.16 The IEEE 118-bus System set 18 – load and wind, 50 scenarios

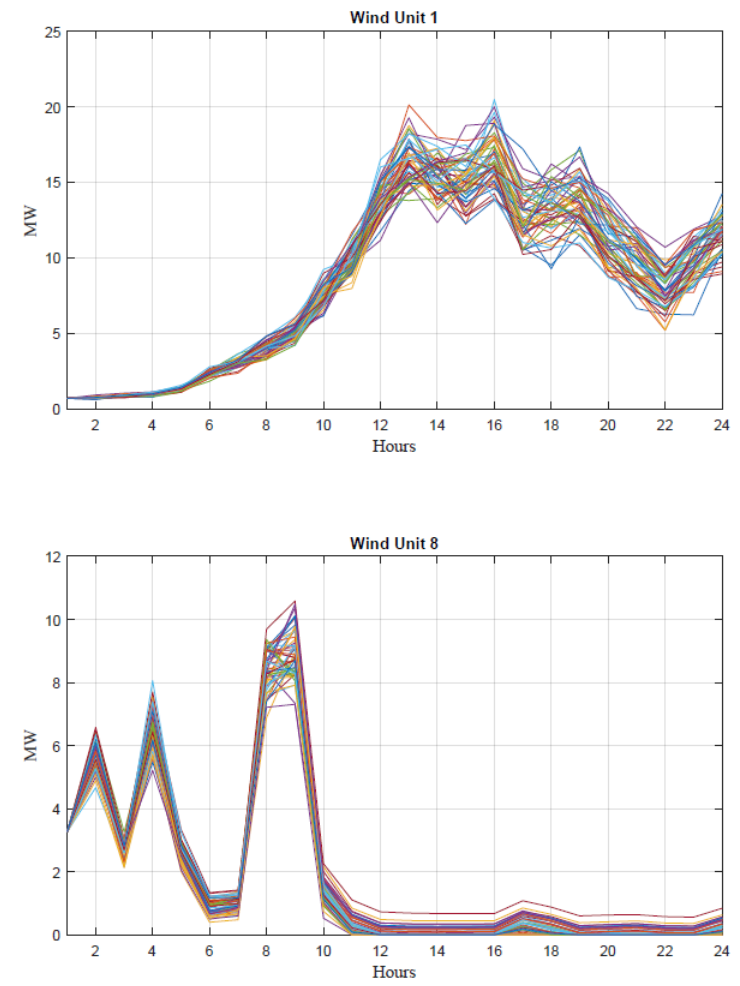


Figure 3.17 The IEEE 118-bus System set 18 – WTG 1 and 8, 50 scenarios

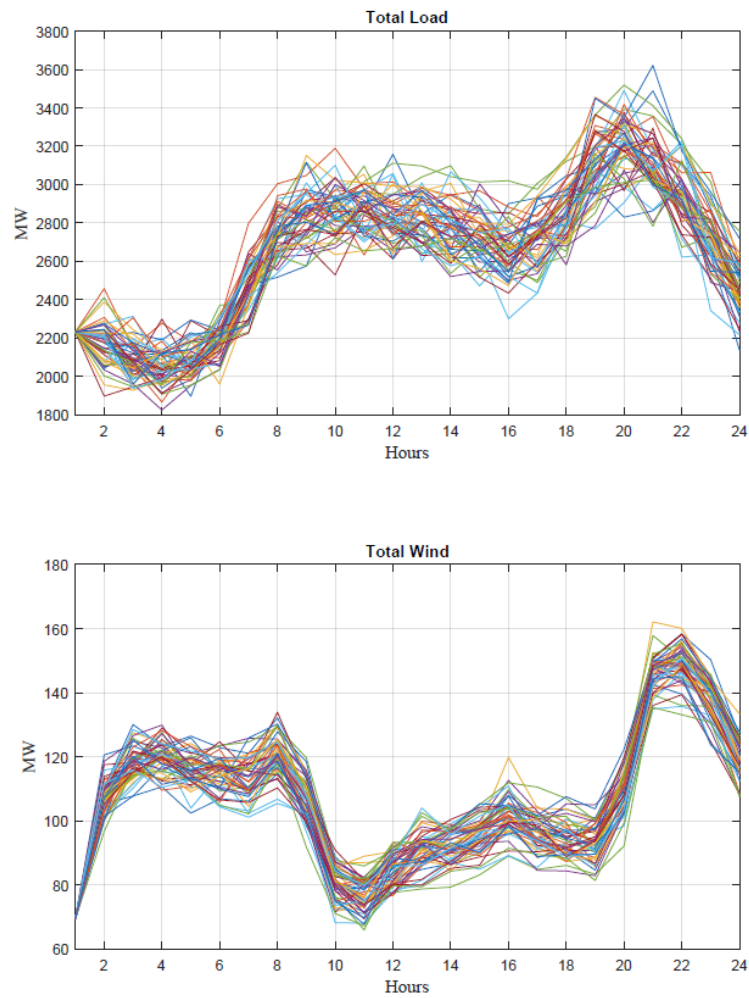


Figure 3.18 The IEEE 118-bus System set 25 - load and wind, 50 scenarios

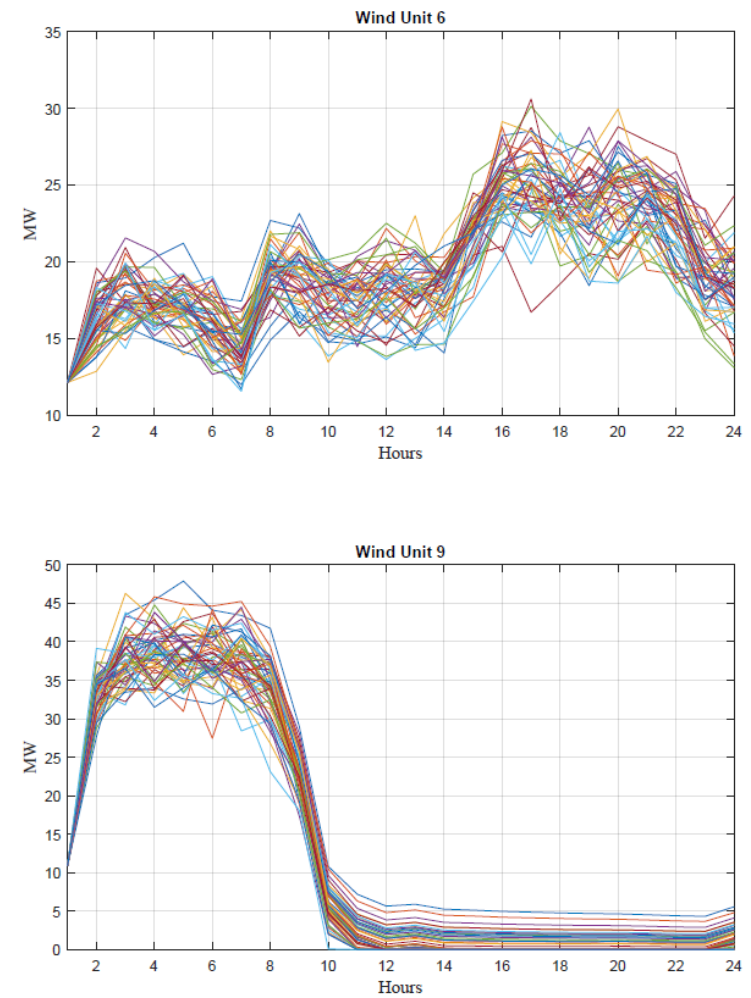


Figure 3.19 The IEEE 118-bus System set 25 – WTG 6 and 9, 50 scenarios

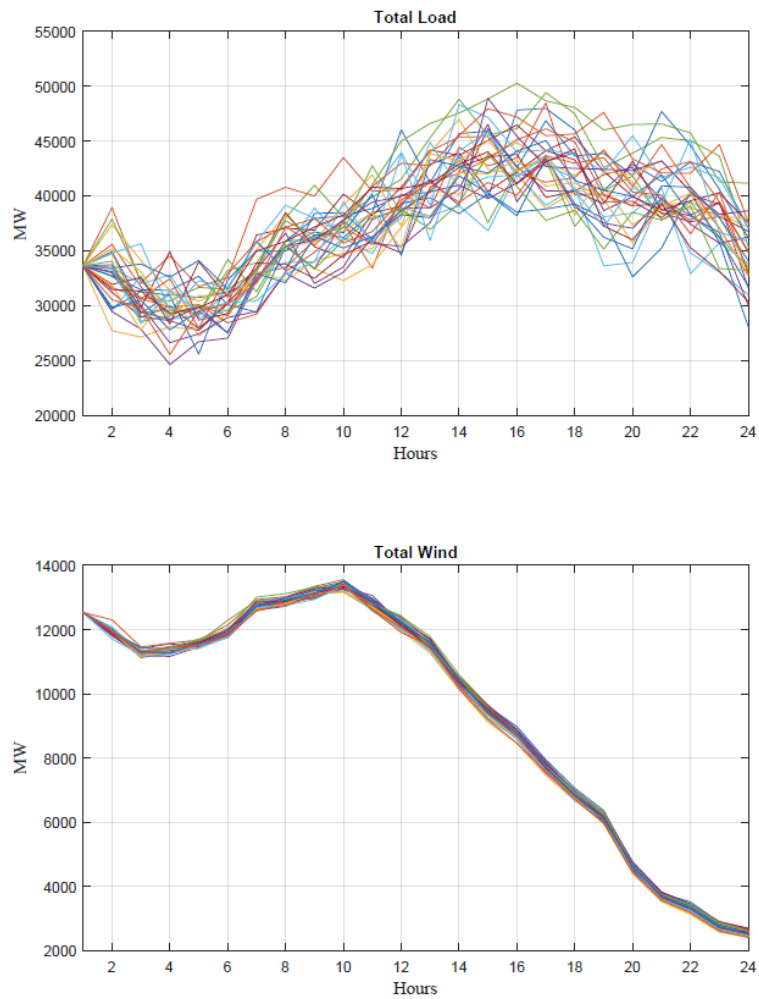


Figure 3.20 The ERCOT-like System – load and wind, 30 scenarios

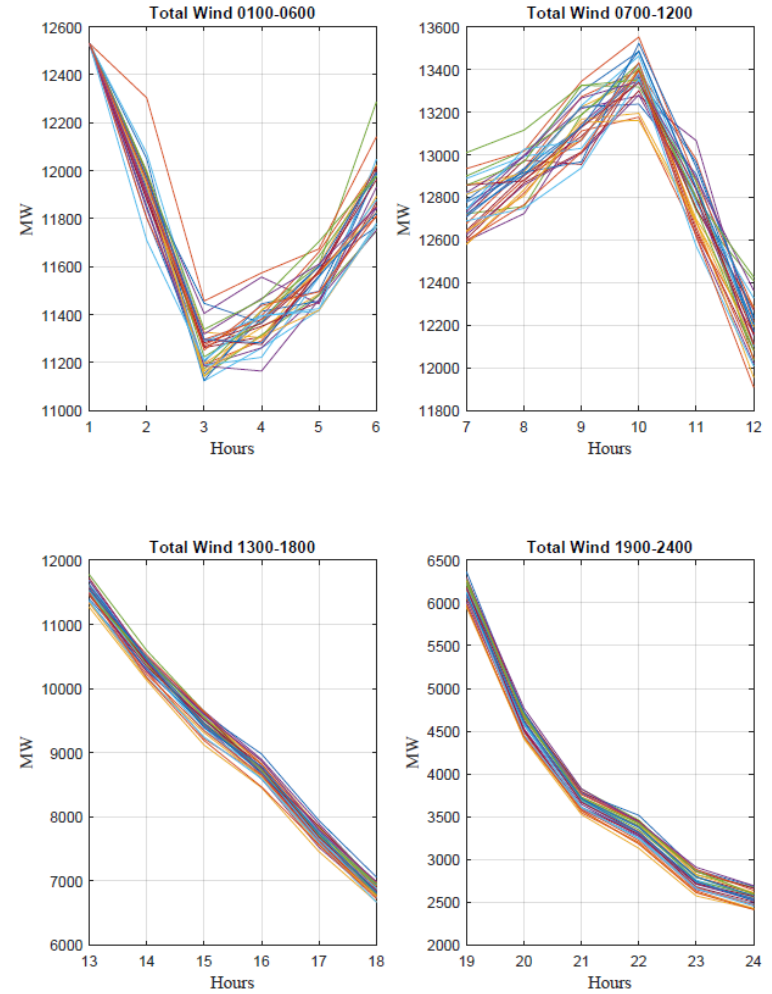


Figure 3.21 The ERCOT-like System – wind, 30 scenarios

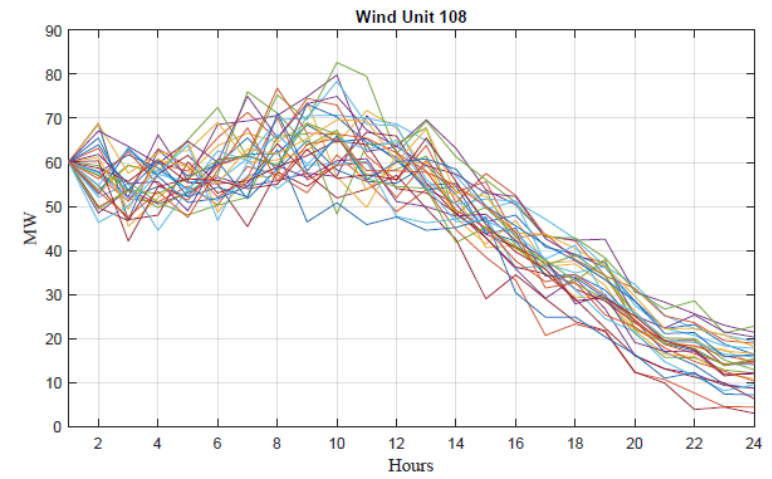
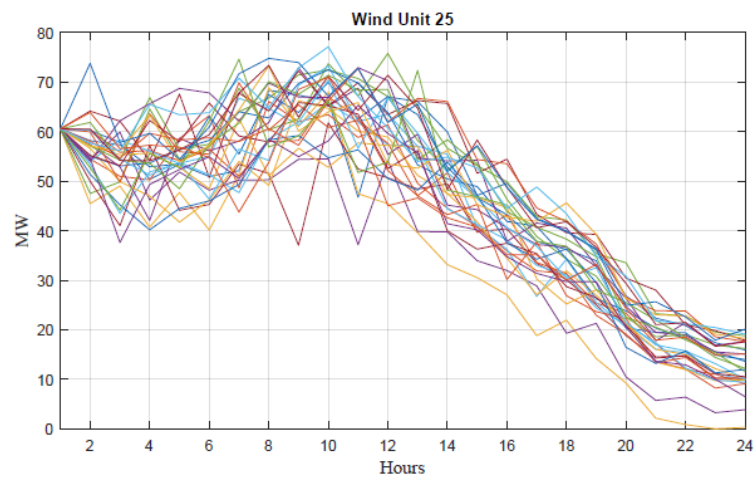
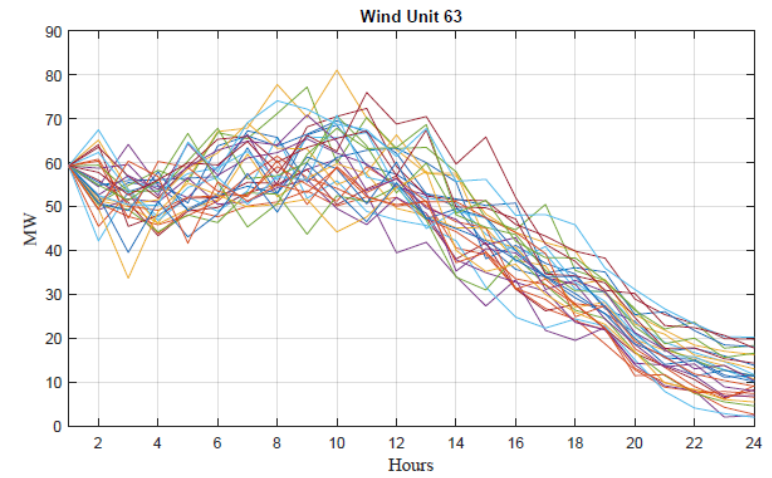
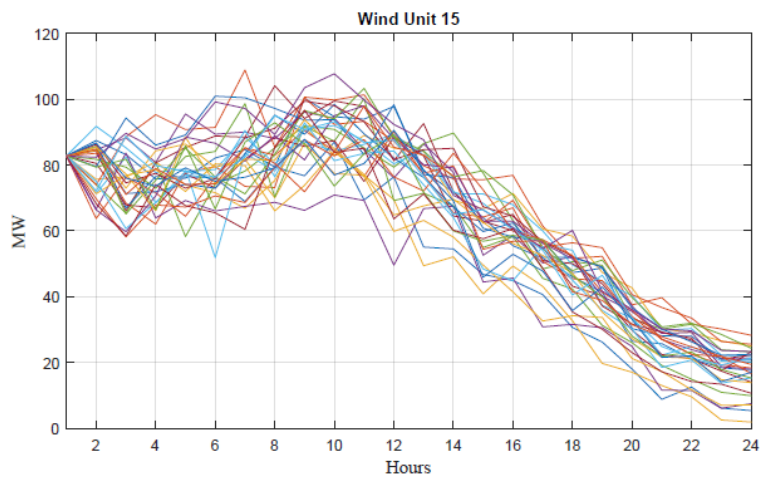


Figure 3.22 The ERCOT-like System – WTG 15 and 25, 30 scenarios

Figure 3.23 The ERCOT-like System – WTG 63 and 108, 30 scenarios

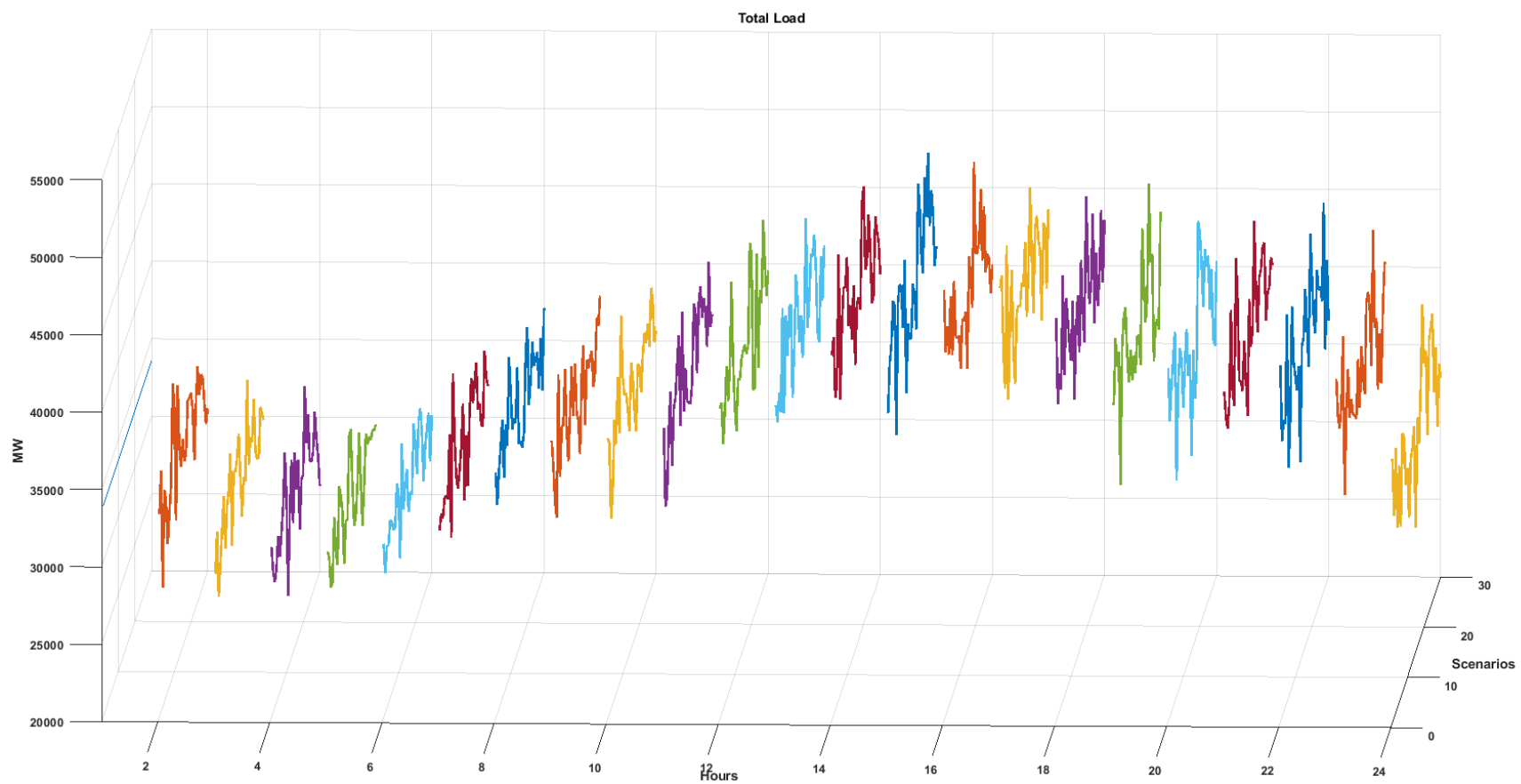


Figure 3.24 The ERCOT-like System total load variation across scenarios for each hour

The quality of the scenarios generated using ARMA seems to be adequate for the case studies in this research. It is apparent from the 25 sets of 50 scenarios that were generated for the IEEE 118 bus System results. Also, the variation used in the ARMA model to generate load scenarios is far higher than the accuracy of the load forecasts in the real world. For example, the generated load forecast error of the ERCOT-like System varies to as high as 45%, while the real-world forecast errors are 2 to 5%. This high variation demonstrates the quality of the solution.

3.6. Extensive Formulation Results

In this section results obtained by solving S-NCUC problem using EF is discussed. EF yielded results for two of the three systems modelled in this research. EF is implemented within the FESTIV environment.

3.6.1. RTS-96 System

The only scenario created for this system, RTS96-10-S1, is solved using EF in the FESTIV environment. The optimization engine used was CPLEX, and the partial output from the CPLEX is shown in Figure 3.25. As shown, the CPLEX took 14 minutes to arrive at a solution. Figure 3.26 shows the FESTIV output where the first stage CPLEX solution is validated for all the ten scenarios. This result shows that all the ten scenarios had neither loss-of-load nor additional load slack, validating the feasibility of the solution obtained.

MIP Solution: 333732.835728 (465363 iterations, 4100 nodes)
 Final Solve: 333732.835728 (758 iterations)

Best possible: 333399.337717
 Absolute gap: 333.498011
 Relative gap: 0.000999

Solution Quality Statistics:

	unscaled		scaled	
	max	sum	max	sum
primal infeasibility	4.619e-14	1.634e-13	2.887e-15	8.671e-14
dual infeasibility	0.000e+00	0.000e+00	0.000e+00	0.000e+00
primal residual	7.550e-15	5.062e-12	7.550e-15	5.062e-12
dual residual	1.421e-14	2.274e-12	1.421e-14	2.274e-12
primal solution vector	1.019e+03	2.773e+04	4.000e+00	2.124e+04
dual solution vector	2.917e+02	1.290e+06	1.024e+03	1.577e+06
slacks	2.400e+01	1.397e+05	2.400e+01	1.175e+05
reduced costs	1.002e+05	2.706e+09	1.002e+05	2.706e+09

Condition number of the scaled basis matrix = 7.657e+02

MIP Kappa distribution Report:

Percentage of stable bases ($\kappa < 1e+7$): 99.88%
 Percentage of suspicious bases ($1e+7 < \kappa < 1e+10$): 0.12%
 Percentage of unstable bases ($1e+10 < \kappa < 1e+14$): 0.00%
 Percentage of illposed bases ($1e+14 < \kappa$): 0.00%
 Max condition number: 2.6849e+07
 Attention index (if > 0.03 caution is advised) 0.00

--- Reading solution for model SCUC
 --- EF.gms(1695) 78 Mb 858 secs
 --- Executing after solve: elapsed 0:14:21.862
 --- EF.gms(1728) 78 Mb
 --- GDX File D:\PSOC\APalani\FESTIV_EF\TOTAL_DASCUCOUTPUT_EF.gdx
 *** Status: Normal completion
 --- Job EF.gms Stop 12/29/19 17:32:24 elapsed 0:14:21.958

Figure 3.25 Partial output of EF CPLEX solution for RTS96-10-S1


```

>> FESTIV
*****
* Flexible Energy Scheduling Tool for Integrating Variable generation *
*****
** FFFFFFF EEEEE SSSSS TTTTTT IIIII VV VV **
** FF EE SS TT II VV VV **
** FFFFFFF EEEEE SSSSS TT II VV VV **
** FF EE SS TT II VV **
** FF EEEEE SSSSS TT IIIII V **
*****
***** National Renewable Energy Laboratory *****
*****

Input File: RTS96_July.h5
Reading Input Files...Complete! (00 min, 05.72 s)
Modeling Initial Day-Ahead Unit Commitment...
Computation Start Time 2019-09-24 18:40:18.2

Production Cost 333732.835728

Computation Stop Time 2019-09-24 18:54:32.9

Scenario 1, PRODCOST 330890.239681, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 2, PRODCOST 331472.473396, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 3, PRODCOST 336667.769211, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 4, PRODCOST 337440.288831, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 5, PRODCOST 331662.192471, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 6, PRODCOST 322261.970781, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 7, PRODCOST 337682.355398, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 8, PRODCOST 337208.446790, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 9, PRODCOST 333360.172644, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Scenario 10, PRODCOST 338682.448077, loss_load 0.000000, add_load_slack 0.000000, startup cost 6497.000000
Gap = 0.000000, Percentage Gap = 0.000000241 fprintf("\n*****");
K>>

```

Figure 3.26 FESTIV output of EF solution for scenario RTS96-10-S1

This result will be used in subsequent chapters for comparison with proposed algorithm.

3.6.2. The IEEE 118-bus

For this system, EF solutions of two of the 27 sets of scenarios created are presented. Results for other sets will be presented in Chapter 5 when Out-of-Sample scenarios are discussed. The scenario sets presented here are:

- IEEE118-10-S1
- IEEE118-50-S0

Figures 3.27 and 3.28 show EF results for 10 scenarios of the IEEE 118-bus System while figures 3.29 and 3.30 show the EF results for the 50 scenarios.

MIP Solution: 851152.261098 (44409 iterations, 172 nodes)
 Final Solve: 851152.260974 (933 iterations)

Best possible: 850355.362209
 Absolute gap: 796.898889
 Relative gap: 0.000936

Solution Quality Statistics:

	unscaled		scaled	
	max	sum	max	sum
primal infeasibility	3.638e-12	3.638e-12	3.638e-12	3.638e-12
dual infeasibility	0.000e+00	0.000e+00	0.000e+00	0.000e+00
primal residual	5.678e-13	2.694e-11	5.596e-14	2.694e-11
dual residual	2.274e-13	1.243e-11	2.274e-13	1.243e-11
primal solution vector	1.040e+04	7.232e+04	8.000e+00	4.799e+04
dual solution vector	1.552e+03	8.266e+06	2.719e+06	1.142e+08
slacks	6.001e+03	3.436e+07	9.980e+02	7.814e+05
reduced costs	3.983e+07	1.051e+10	3.983e+07	1.052e+10

Condition number of the scaled basis matrix = 2.357e+05

MIP Kappa distribution Report:

Percentage of stable bases ($\kappa < 1e+7$): 100.00%
 Percentage of suspicious bases ($1e+7 < \kappa < 1e+10$): 0.00%
 Percentage of unstable bases ($1e+10 < \kappa < 1e+14$): 0.00%
 Percentage of illposed bases ($1e+14 < \kappa$): 0.00%
 Max condition number: 7.8511e+06
 Attention index (if >0.03 caution is advised) 0.00

--- Reading solution for model SCUC

--- EF.gms(1724) 592 Mb

--- GDX File D:\PSOC\APalani\FESTIV_EF\TEMP\TOTAL_DASCUCOUTPUT_EF.gdx

*** Status: Normal completion

--- Job EF.gms Stop 12/20/19 13:46:23 elapsed 0:11:33.465

Figure 3.27 Partial output of EF CPLEX solution for IEEE118-10-S1

```

>> FESTIV
*****
* Flexible Energy Scheduling Tool for Integrating Variable generation *
*****
** FFFFFFFF EEEEEEE SSSSSS TTTTTTTT IIIIII VV VV **
** FF EE SS TT II VV VV **
** FFFFFFFF EEEEEEE SSSSSS TT II VV VV **
** FF EE SS TT II VV **
** FF EEEEEEE SSSSSS TT IIIIII V **
*****
***** National Renewable Energy Laboratory *****
*****

Input File: IIT_118bus_CAISO_basecase_err00_Jan.h5
Reading Input Files...Complete! (00 min, 11.24 s)
Modeling Initial Day-Ahead Unit Commitment...
Computation Start Time 2019-09-22 22:27:51.4

Production Cost 851152.260974

Computation Stop Time 2019-09-22 22:42:31.7

Scenario 1, PRODCOST 858742.951559, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 2, PRODCOST 899517.065818, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 3, PRODCOST 879967.170613, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 4, PRODCOST 913539.710146, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 5, PRODCOST 933002.900434, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 6, PRODCOST 822549.270427, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 7, PRODCOST 811774.894402, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 8, PRODCOST 736196.646091, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 9, PRODCOST 754780.653647, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Scenario 10, PRODCOST 901451.347839, loss_load 0.000000, add_load_slack 0.000000, startup cost 24337.200000
Gap = 0.000124, Percentage Gap = 0.000000241 fprintf("\n*****");
K>>

```

Figure 3.28 FESTIV output of EF solution for scenario IEEE118-10-S1

MIP Solution: 886970.463993 (317863 iterations, 260 nodes)
 Final Solve: 886970.463872 (6325 iterations)

Best possible: 885635.249893
 Absolute gap: 1335.214100
 Relative gap: 0.001505

Solution Quality Statistics:

	unscaled		scaled	
	max	sum	max	sum
primal infeasibility	3.638e-12	3.638e-12	3.638e-12	3.638e-12
dual infeasibility	9.142e-07	1.429e-05	9.142e-07	1.429e-05
primal residual	6.085e-13	1.356e-10	5.596e-14	1.356e-10
dual residual	2.674e-13	1.858e-11	2.674e-13	1.858e-11
primal solution vector	3.390e+03	2.573e+05	8.604e+00	2.373e+05
dual solution vector	1.410e+03	8.351e+06	5.438e+05	1.142e+08
slacks	6.001e+03	1.688e+08	9.980e+02	3.856e+06
reduced costs	3.983e+07	1.051e+10	3.983e+07	1.052e+10

MIP Kappa distribution Report:

Percentage of stable bases ($\kappa < 1e+7$): 100.00%
 Percentage of suspicious bases ($1e+7 < \kappa < 1e+10$): 0.00%
 Percentage of unstable bases ($1e+10 < \kappa < 1e+14$): 0.00%
 Percentage of illposed bases ($1e+14 < \kappa$): 0.00%
 Max condition number: 2.4099e+06
 Attention index (if > 0.03 caution is advised) 0.00

--- Reading solution for model SCUC
 --- EF.gms(1728) 2972 Mb
 --- GDX File D:\PSOC\APalani\FESTIV_EF\TEMP\TOTAL_DASCUCOUTPUT_EF.gdx
 *** Status: Normal completion
 --- Job EF.gms Stop 07/20/20 20:51:16 elapsed 5:03:56.063

Figure 3.29 Partial output of EF CPLEX solution for IEEE118-50-S0

```

>> FESTIV
*****
* Flexible Energy Scheduling Tool for Integrating Variable generation *
*****
** FFFFFF EEEEE SSSSSS TTTTTTT IIIIII VV VV **
** FF EE SS TT II VV VV **
** FFFFFFF EEEEE SSSSSS TT II VV VV **
** FF EE SS TT II VV **
** FF EEEEE SSSSSS TT IIIIII V **
*****
***** National Renewable Energy Laboratory *****
*****

Input File: IIT_118bus_CAISO_basecase_err00_Jan.h5
Reading Input Files...Complete! (00 min, 09.48 s)
Modeling Initial Day-Ahead Unit Commitment...
Computation Start Time 2020-07-20 15:47:17.2

Production Cost 886970.463872

Computation Stop Time 2020-07-20 20:51:17.2

Scenario 1, PRODCOST 776860.965492, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 2, PRODCOST 883371.569919, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 3, PRODCOST 946136.805107, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 4, PRODCOST 952932.334293, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 5, PRODCOST 840632.672279, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 6, PRODCOST 881175.101386, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 7, PRODCOST 897764.980499, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 8, PRODCOST 801937.461417, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 9, PRODCOST 932687.165653, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 10, PRODCOST 852732.053994, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 11, PRODCOST 1055224.264343, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 12, PRODCOST 866901.140118, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 13, PRODCOST 910213.384257, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 14, PRODCOST 762923.865358, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 15, PRODCOST 879641.452410, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 16, PRODCOST 946380.564789, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 17, PRODCOST 987475.822744, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 18, PRODCOST 1009116.177746, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 19, PRODCOST 1038437.780394, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 20, PRODCOST 796511.402666, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 21, PRODCOST 867687.920093, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 22, PRODCOST 812555.719244, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 23, PRODCOST 805026.882632, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 24, PRODCOST 953843.876697, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 25, PRODCOST 966086.682182, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 26, PRODCOST 831907.250084, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 27, PRODCOST 881852.296517, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 28, PRODCOST 847445.476249, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 29, PRODCOST 900734.108811, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 30, PRODCOST 891657.355859, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 31, PRODCOST 846633.065210, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 32, PRODCOST 847384.897147, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 33, PRODCOST 785546.272760, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 34, PRODCOST 862848.256793, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 35, PRODCOST 911529.979089, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 36, PRODCOST 975059.681516, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 37, PRODCOST 955982.768013, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 38, PRODCOST 983288.720076, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 39, PRODCOST 812077.655228, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 40, PRODCOST 842412.587004, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 41, PRODCOST 892486.843422, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 42, PRODCOST 781589.725264, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 43, PRODCOST 908799.325001, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 44, PRODCOST 901533.180087, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 45, PRODCOST 871262.828227, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 46, PRODCOST 856284.799721, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 47, PRODCOST 896877.139234, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 48, PRODCOST 844228.716716, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 49, PRODCOST 803047.556003, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Scenario 50, PRODCOST 991794.669652, loss_load 0.000000, add_load_slack 0.000000, startup cost 19951.200000
Gap = 0.000116, Percentage Gap = 0.000000

```

Figure 3.30 FESTIV output of EF solution for scenario IEEE118-50-S0

Comparing the time taken to obtain EF solution for ten scenarios with 50 scenarios shows how the time taken increases exponentially with the increased scenarios. The time taken for ten scenarios is 11 minutes, while the time taken for 50 scenarios is 303 minutes. Also, note that the EF solution obtained for each set is feasible for every scenario within the respective set.

3.6.3. The ERCOT-like Large System

Unable to obtain a solution within the computational constraints for this extensive system using EF. This is why the scenario-wise decomposition method is needed, and a discussion of such a method will be discussed in the next chapter.

3.7. Summary

This chapter started with a brief description of Unit Commitment followed by the formulation of Stochastic Network Constrained Unit Commitment. The computational environment, along with systems employed in the simulations, were discussed. The systems on which simulations conducted are small, medium, and large real-world-like systems. These systems vary in size from 10s of nodes to 1000s of nodes. An integral part of stochastic programming – scenario generation – was discussed, and several scenarios generated using ARMA were depicted. Finally, the deterministic version of S-NCUC solved using EF was presented for two of the three systems considered. EF was not tractable for the extensive ERCOT-like system. Obtaining result for such a large system is the discussion of the next chapter.

4. Fast Penalty-Based Gauss-Seidel Algorithm

The computational complexity of the Extensive Formulation (EF) used in Chapter 3 to solve the S-NCUC problem grows exponentially with the size of the system and the number of scenarios. This deterministic formulation leads to computationally intractable situations even for a medium system with a reasonable number of scenarios. This is why a decomposition-based algorithm is used in solving S-NCUC. This chapter starts with a discussion on the decomposition using augmented Lagrangian, followed by the formulation of the S-NCUC problem for the application of the Penalty-Based Gauss-Seidel (PBGS) algorithm. Presentation of S-NCUC results obtained using PBGS is presented. Innovation to PBGS algorithm called Fast PBGS is developed, and the time saved by Fast PBGS is compared with PBGS at the end of this chapter.

4.1. Augmented Lagrangian

The scenario-based decomposition framework separates the first-stage S-NCUC into each individual scenario (i.e., $I_{i,t}^s$) and introduces a coupling constraint that enforces the first-stage S-NCUC decisions of non-quick-start generators (NQG) to be the same across all scenarios. This constraint for NQGs, termed as a non-anticipativity constraint (NAC), is given below.

$$I_{i,t} = I_{i,t}^s = Z_{i,t}, \quad \forall i \in NI, \forall t \in NT, \forall s \in S \quad (4.1)$$

As UCs of quick-start generators (QSGs) can vary across the scenarios considered, constraint (4.1) is not imposed on quick-start units. By relaxing constraint (4.1), the augmented Lagrangian function is defined as

$$\varphi^+(\mathbf{I}, \mathbf{p}, \boldsymbol{\omega}, \boldsymbol{\rho}) =$$

$$\begin{aligned} \underset{I, IU, p}{\text{minimize}} \sum_{s=1}^S Pr^s & \left(\sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left(NL_i \cdot I_{i,t}^s + SU_{i,t} \cdot IU_{i,t}^s + \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) + \right. \right. \right. \\ & \left. \left. \left. \omega_{i,t}^s \cdot (I_{i,t}^s - Z_{i,t}) + \alpha \cdot \psi_{\rho}^s(I_{i,t}^s - Z_{i,t}) \right) \right. \right. \\ & \left. \left. + (AL_t^s + LL_t^s) \cdot VOLL \quad + \sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right] \right) \end{aligned} \quad (4.2)$$

where,

$$\boldsymbol{\omega}, \boldsymbol{\rho} \in \mathbb{R}^{NI \times NT \times S}$$

$\psi_{\rho}^s(I_{i,t}^s - Z_{i,t})$, denoted as $\psi_{\rho}^s(\cdot)$, is an augmenting function associated with NAC in (4.1) and a penalty factor $\boldsymbol{\rho}$; α is a scalar.

Enforcing the probability-weighted sum of the dual variable ($\sum_{s \in S} Pr^s \boldsymbol{\omega}^s = 0$) is a necessary condition for the augmented Lagrangian function (4.2) to be bounded from below. The NAC defines a subspace \mathcal{N} and the optimality conditions require that the dual variable ($\boldsymbol{\omega}$) lie in the subspace of \mathcal{N}^{\perp} [51]. This requirement, $\sum_{s \in S} Pr^s \boldsymbol{\omega}^s = 0$ is enforced by the updating of the $\boldsymbol{\omega}$ in every iteration of PHA algorithm. Proof of this condition is given in [85].

Under this imposed condition, $Z_{i,t}$ associated with $\omega_{i,t}^s$ in (4.2) vanishes. Also, $Z_{i,t} \cdot \omega_{i,t}^s$ is a constant added to the objective function and does not need to be in the objective function. The augmented Lagrangian function (4.2) is decomposed into S subproblems, each representing an individual-scenario NCUC problem to be solved. Two algorithms are considered in this research, namely PBGS and FW-PHA, which solve the augmented Lagrangian function (4.2) using different augmenting functions $\psi_{\rho}(\cdot)$.

4.2. Strong Duality with Augmented Lagrangian

Achieving strong duality (zero duality gap) helps imply convergence, and more importantly the solution obtained is a primal solution. The convergence is defined as when all the individual scenarios first stage decision $I_{i,t}$ agrees with implementable $Z_{i,t}$, i.e., equation (4.1) is satisfied. Only then a solution - that does not require any further processing before it is implemented - is found. The presence of integers makes UC problem non-convex. This non-convexity is the cause of non-zero duality gap when Lagrangian relaxation is used. Augmented Lagrangian is a modification of classical Lagrangian with an augmenting function with two parts; it is a function multiplied by a coefficient. Both the augmenting function for $\psi_\rho(\cdot)$ and the coefficient called penalty factor ρ will be discussed later in the chapter.

4.2.1. Affine Functions

An Affine Function is a linear function that has non-zero constant ($\mathbf{Ax} + \mathbf{b}$). In the non-convex problems like UC, use of affine function as $\psi_\rho(\cdot)$ might not close the gap (the supporting epigraph of the primal function using hyperplanes does not work [86]). This is explained in detail in Chapter 11, section K of [87]. This is apparent in Figure 3.1 which is reproduced from [87]. In this figure the affine augmenting function ψ is prevented from getting to the value of the objective function φ . It is to be noted that strong duality occurs at $\inf \varphi(\cdot) = \sup \psi(\cdot)$. In Figure 4.1, there is a gap between these two functions.

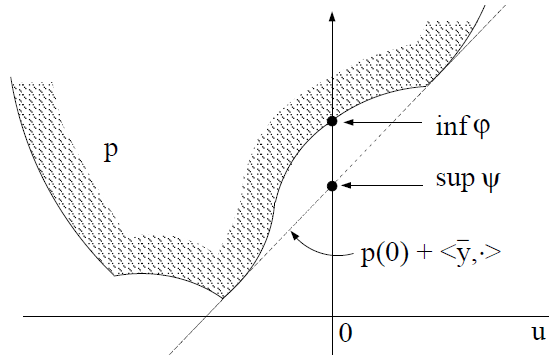


Figure 4.1 Duality gap in minimization problems lacking adequate convexity [87]

4.2.2. Non-linear Functions

Figure 4.1 suggests that perhaps a function such as quadratic function can penetrate such ‘dents’ of φ (epigraph of the primal function can be supported by non-linear surfaces). This is shown in Figure 4.2 which is also reproduced from [87]. Wang *et.al.*, showed, under certain conditions, that the strong duality can be achieved asymptotically by using non-linear augmented Lagrangian function and increasing the associated penalty factor ρ to infinity [88].

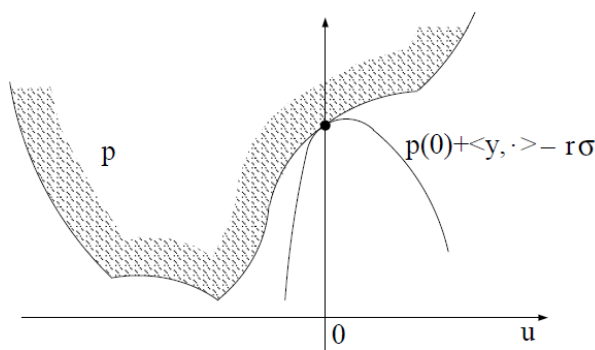


Figure 4.2 Duality gap removed by an augmenting function [87]

Even the non-linear augmented Lagrangian cannot yield strong duality for some class of non-linear optimization problems. This has been proven mathematically by Feizollahi *et.al.* through an example [89]. Feizollahi showed that only augmented Lagrangian using sharp functions can penetrate certain ‘dents’ in certain types of non-convex problems [89], [90]. This is shown in

Figure 4.3 where only $\psi_3(u)$ can penetrate and reach $p(u)$ to achieve strong duality while the other two functions $\psi_1(u)$ and $\psi_2(u)$ are unable to do so.

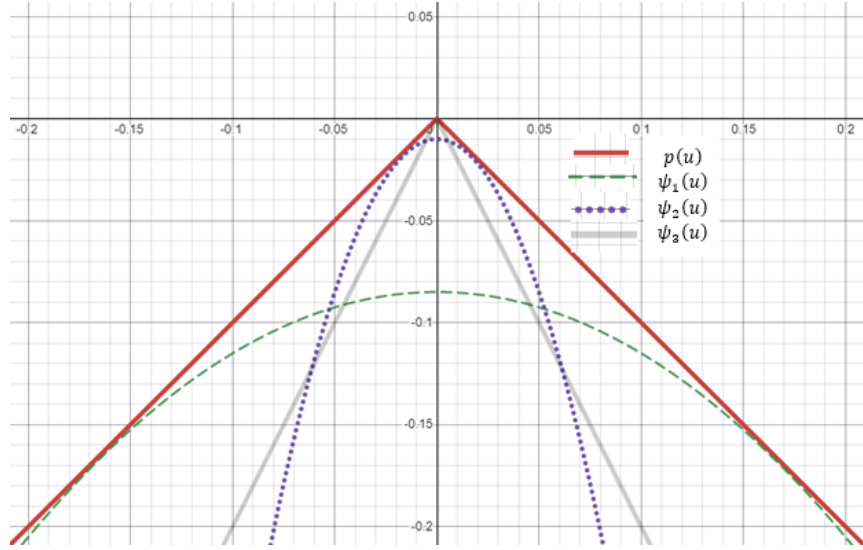


Figure 4.3 Value function and some augmenting functions [90]

4.3. Exact Augmented Lagrangian

The function that achieves a strong duality gap with a finite value of penalty factor is known as ‘Exact Penalization’ or ‘Exact’. Burke, in the abstract of [91], mentions that Eremin (in 1966) and Zangwill (in 1967) introduced a notion of exact penalization for use in the development of algorithms for constrained optimization. Later, there have been several publications in this area in general [92]-[98], and to MIP in particular [89], [99] and [100].

An Exact augmented Lagrangian is a class of exact penalty methods whose objective is to solve a constrained optimization (primal) problem through an unconstrained optimization problem that has the same local (global) solutions as the primal problem. Often, in a non-convex problem like S-NCUC, only the local minimum is found. This local minimum is an exact augmented Lagrangian solution and is referred to as an *exact solution* hereinafter.

Definition 1 (Exact penalty representation [89]): A given augmenting function $\psi_\rho(\cdot)$ in $\varphi^+(\cdot)$ is an exact penalty representation, if there exists a dual vector ω , for all ρ sufficiently large, such that:

$$\zeta = \varphi^+ \quad (4.3)$$

and,

$$\underset{I,p}{\operatorname{argmin}} \zeta = \underset{I,p}{\operatorname{argmin}} \varphi^+(I, p, \omega, \rho) \quad (4.4)$$

Definition 1 indicates that, with an exact augmenting function $\psi_\rho(\cdot)$, an optimal solution to the augmented Lagrangian $\varphi^+(\cdot)$ is also a (local) optimal solution to the extensive form ζ (3.1) of S-NCUC. The exactness implies that the optimal Lagrangian dual solution to $\varphi^+(\cdot)$ is directly primal feasible with all the relaxed NAC constraints being satisfied. Hence, no additional algorithm is required to restore the primal feasibility from a Lagrangian dual solution. This constitutes one salient merit of using an exact augmenting function, which will be demonstrated in comparison with PHA in Chapter 6. It is worth mentioning that Definition 1 emphasizes the feasibility rather than the optimality of a solution to augmented Lagrangian $\varphi^+(\cdot)$. Specifically, an optimal solution to $\varphi^+(\cdot)$ is a local minimum to the extensive form ζ due to the non-convexity of S-NCUC [97]. Therefore, a quantitative assessment of the quality of a solution to $\varphi^+(\cdot)$ is needed. Discussion on such an assessment is presented in Chapter 5.

4.4. Penalty-Based Gauss-Seidel Algorithm

An important question is how to construct an exact augmenting function ψ_ρ^s . The discussion and work here is based on the findings of Oliveira *et.al.*, in [101]. An l_1 norm-like function based on a semi-Lagrangian approach is employed. Unlike the semi-Lagrangian approach - wherein an equality constraint is reformulated as a pair of inequality constraints and the Lagrangian relaxation - is applied to either of the pair [102], the NAC (4.1) here is reformulated as two inequality constraints and then relax both constraints using two penalty factors $(\underline{\rho}, \bar{\rho})$ as follows:

$$\psi_\rho^s(I_{i,t}^s - Z_{i,t}) := \underline{\rho}_{i,t}^s \cdot [I_{i,t}^s - Z_{i,t}]^- + \bar{\rho}_{i,t}^s \cdot [Z_{i,t} - I_{i,t}^s]^- \quad (4.5)$$

where $[\cdot]^-$ represents the positive basis, which is defined as $-\min\{0, \cdot\}$;

Equation (4.5) can be implemented as follows:

$$\psi_\rho^s(I_{i,t}^s - Z_{i,t}) := \underline{\rho}_{i,t}^s \cdot \underline{u} + \bar{\rho}_{i,t}^s \cdot \bar{u} \quad (4.6)$$

$$\text{Such that: } \underline{u} \geq 0, \underline{u} \geq (Z_{i,t} - I_{i,t}^s)$$

$$\bar{u} \geq 0, \bar{u} \geq (I_{i,t}^s - Z_{i,t})$$

The augmenting function (4.5) is an exact penalty representation since it meets the following three criteria:

- $\psi_\rho^s(0) = 0$
- $\psi_\rho^s(I_{i,t}^s - Z_{i,t}) \geq \delta > 0, \forall (I_{i,t}^s - Z_{i,t}) \notin V$
- $\psi_\rho^s(I_{i,t}^s - Z_{i,t}) \geq \nu \|I_{i,t}^s - Z_{i,t}\|_\infty, \forall (I_{i,t}^s - Z_{i,t}) \in V$

for some open neighborhood V of 0 and positive scalars $\delta, \nu > 0$. The exactness proof of augmenting function (4.5) is given in Theorem 5 of [89].

It is worthwhile noting that a common choice for the augmenting function $\psi_\rho(\cdot)$ in the literature is the use of norms. For example, the PHA algorithm uses the square of l_2 norm, i.e., $\frac{\rho}{2} \|I_{i,t}^s - Z_{i,t}\|_2^2$ as the augmenting function (proximal Lagrangian) [51]. As indicated in the third criterion above, the value of the augmenting function must be greater than or equal to an infinity norm in the neighborhood of zero. The PHA algorithm does not satisfy this criterion and therefore is not an exact penalty representation for MIP problem like S-NCUC. This is depicted in Figure 4.3. The augmenting function in (4.5) is a sharp Lagrangian and under any one of the following conditions, ω can be set to $\mathbf{0}$.

Condition 1: Proposition 8 in [89] states that the augmenting function using any norm, for any $\omega \in \mathbb{R}^{NI \times NT \times S}$ there exists a finite ρ such that the augmented Lagrangian solution is the same as the primal MIP solution.

Condition 2: Definition 8 in [89] and 11.60-11.62 in [87] states that in the sharp Lagrangian, suppose that $\varphi^+(\mathbf{0}) > -\infty$ for some $\rho \in (0, \infty)$. Then, a necessary and sufficient condition for the vector $\omega = \mathbf{0}$ to support an exact penalty representation is that the value function $p(u)$ is calm from below at $u = \mathbf{0}$.

Therefore, setting $\omega = \mathbf{0}$ and omitting ω from subsequent discussions is done for the simplicity of representation.

The second important consideration is in the calculation of implementable $Z_{i,t}$. Since UC takes a binary value, it is imperative to maintain $Z_{i,t}$ binary. Calculation of $Z_{i,t}$ can be accomplished by minimizing (4.6) with respect to $Z_{i,t}$ using fixed $I_{i,t}^s$ over all the scenarios.

$$\begin{aligned} \mathbf{Z} &:= \underset{\mathbf{Z}}{\operatorname{argmin}} \sum_{s \in S} (\underline{\rho}_{i,t}^s \cdot \underline{u} + \bar{\rho}_{i,t}^s \cdot \bar{u}) \\ \text{Such that: } \underline{u} &\geq 0, \underline{u} \geq (Z_{i,t} - I_{i,t}^s) \\ \bar{u} &\geq 0, \bar{u} \geq (I_{i,t}^s - Z_{i,t}) \end{aligned} \quad (4.7)$$

However, if the problem is restricted to take only binary values, the $Z_{i,t}$ can be calculated as follows at every iteration [101]:

$$Z_{i,t} := \begin{cases} 1, & \text{if } \sum_{s \in S} (1 - I_{i,t}^s) \cdot \underline{\rho}_{i,t}^s < \sum_{s \in S} I_{i,t}^s \cdot \bar{\rho}_{i,t}^s \\ 0, & \text{if } \sum_{s \in S} (1 - I_{i,t}^s) \cdot \underline{\rho}_{i,t}^s > \sum_{s \in S} I_{i,t}^s \cdot \bar{\rho}_{i,t}^s \\ 0 \text{ or } 1, & \text{otherwise} \end{cases} \quad (4.8)$$

As shown, the implementable takes a UC consensus on the majority of scenarios weighted by the penalty factors, leading to a binary value. In contrast, the PHA algorithm calculates the implementable via a weighted average of UC, resulting in a fractional value that requires further conversion to a binary value.

After $Z_{i,t}$ is obtained, ρ at the k^{th} iteration is updated as:

$$\begin{aligned} \underline{\rho}_{i,t,k}^s &:= \underline{\rho}_{i,t,k-1}^s + \gamma \cdot [I_{i,t,k}^s - Z_{i,t,k}]^- \quad \forall s \in S \\ \bar{\rho}_{i,t,k}^s &:= \bar{\rho}_{i,t,k-1}^s + \gamma \cdot [Z_{i,t,k} - I_{i,t,k}^s]^- \quad \forall s \in S \end{aligned} \quad (4.9)$$

In (4.9), the two penalty factors are updated separately based on the direction of the NAC (4.1) violation. Unlike the PHA algorithm, which uses the square of l_2 norm as the augmenting function to penalize NAC violations uniformly, the penalty factor updated in (4.9) offers more granular modifications to satisfy the NAC constraints.

Proposition 14 [101] states the following:

Suppose a set of scenario-dependent solutions $(I^s)_{s \in S}$, where $I^s = (I_i^s)_{i=1,\dots,n}$, are given and $Z = (Z_i)_{i=1,\dots,n}$. For each $i \in \{1, \dots, n\}$ define

$$I^+(Z_i) := \{s \in S \mid I_i^s > Z_i\}$$

$$I^-(Z_i) := \{s \in S \mid I_i^s < Z_i\}$$

$$I^0(Z_i) := \{s \in S \mid I_i^s = Z_i\}$$

Then Z_i solves problem (14) given fixed $(I^s)_{s \in S}$ if and only if

$$\sum_{s \in I^+(Z_i)} \bar{\rho}_i^s - \sum_{s \in I^-(Z_i)} \underline{\rho}_i^s \in \left[- \sum_{s \in I^0(Z_i)} \bar{\rho}_i^s, \sum_{s \in I^0(Z_i)} \underline{\rho}_i^s \right] \quad (4.10)$$

Proof of this proposition is given in [101].

Now, consider the penalty updated in (4.9). Using (4.9) The difference of differences of $\bar{\rho}_i^s, \underline{\rho}_i^s$ between successive iterations can be written as follows:

$$\Delta_i^{k+1} = \gamma \left[\sum_{s \in I^+(Z_i)} [I_{i,t,k}^s - Z_{i,t,k}]^- - \sum_{s \in I^-(Z_i)} [I_{i,t,k}^s - Z_{i,t,k}]^- \right] \quad (4.11)$$

The optimality condition of (4.10) is improved for large $\bar{\rho}_i^s$ and $\underline{\rho}_i^s$ as this makes the target interval on the right-hand side of (4.10) larger. The exponential multiplying factor α accomplishes the gradual increase in the terms Δ_i^{k+1} in an attempt to improve the convergence. It should be noted that, in order to improve the convergence, scalar α is increased exponentially over iterations and defined as follows:

$$\alpha := \beta^{k-1} - 1 \quad (4.12)$$

where $\beta \in (1, 2]$. The PBGS algorithm is shown in Algorithm 1, in which the indices i and t are omitted for brevity.

4.4.1. PBGS Algorithm

Algorithm 1: Penalty-Based Gauss-Seidel Algorithm

```

1 Initialization  $\rho, \gamma, \beta, \epsilon 1, \epsilon 2, l_{max}, \omega$ 
2 foreach  $s \in S$  do
3    $(I^s, p^s) \leftarrow \underset{I^s, p^s}{\operatorname{argmin}} \{ \varphi^+(I^s, p^s, \omega^s, \rho^s = 0) \mid (I^s, p^s) \in \Lambda^s \}$ 
4 end
5 Calculate  $Z^0 \leftarrow \operatorname{round}(\sum_{s \in S} Pr^s \cdot I^s)$ 
6 while  $(\sum_{s \in S} \|I^s - Z\|_2^2 > \epsilon 1)$  do
7   for  $l = 1, \dots, l_{max}$  do
8     foreach  $s \in S$  do
9        $(I^s, p^s, \phi^{s,l}) \leftarrow \underset{I^s, p^s}{\operatorname{argmin}} \{ \varphi^+(I^s, p^s, \omega^s, \rho^s) \mid (I^s, p^s) \in \Lambda^s \}$ 
10    end
11    Calculate  $Z^l$  using (4.8)
12     $\Gamma \leftarrow \phi^l - \phi^{l-1}$ 
13    if  $(|\Gamma| \leq \epsilon 2)$  or  $(l = l_{max})$  then
14       $Z \leftarrow Z^l$ 
15      break
16    end
17     $l \leftarrow l + 1$ 
18  end
19  Update  $\underline{\rho}^s$  and  $\overline{\rho}^s$  using (4.9)
20  Update  $\alpha$  using (4.12)
21 end
22 return  $Z$ 

```

4.4.2. Proof of convergence of PBGS

Augmented Lagrangian (4.2) in abbreviated form is given as follows. All the constraints (3.2) - (3.7) are applicable.

$$\varphi^+(\mathbf{I}, \mathbf{p}, \boldsymbol{\rho}) = \mathcal{C}(\mathbf{I}, \mathbf{p}) + \boldsymbol{\rho}\psi(\mathbf{I} - \mathbf{Z}) \quad (4.13)$$

where $\mathcal{C}(\mathbf{I}, \mathbf{p}, \boldsymbol{\rho})$ is the scenario-wise decomposed NCUC objective function and $\boldsymbol{\rho}\psi(\mathbf{I} - \mathbf{Z})$ is the augmented Lagrangian term.

Proof 1: The PBGS algorithm uses an exact augmenting Lagrangian. According to Theorem 5 in [89], there exists a finite $\boldsymbol{\rho}^* > \mathbf{0}$ such that for any $\boldsymbol{\rho} \in [\boldsymbol{\rho}^*, \infty)$, the following equation holds regarding (A1).

$$\varphi^+(\mathbf{I}, \mathbf{p}, \boldsymbol{\rho}) = \mathcal{C}(\mathbf{I}, \mathbf{p}) + \boldsymbol{\rho}\psi(\mathbf{I} - \mathbf{Z}) = \mathcal{C}(\mathbf{I}, \mathbf{p}) \quad (4.14)$$

Equation (3.18) can be only possible if and only if

$$\boldsymbol{\rho}\psi(\mathbf{I} - \mathbf{Z}) = 0 \quad (4.15)$$

Equation (4.15) implies that $\mathbf{I} = \mathbf{Z}$, fulfilling the termination criteria in Line 21 of the PBGS algorithm. Due to monotonically increasing $\boldsymbol{\rho}$, $\boldsymbol{\rho} \in [\boldsymbol{\rho}^*, \infty)$ is satisfied within a finite number of iterations. ■

4.4.3. PBGS Results

In this subsection, the quality of the proposed Fast PBGS solutions is evaluated by comparing with the EF solutions obtained in Chapter 3. Like in the Chapter 3 only a small subset of the scenarios is compared here. In Chapter 5 all scenarios will be presented.

All gap and difference calculations are based on the following equation, where φ_x represents other methods with which the PBGS is compared against, such as EF, FW-PHA, or PHA [103]. The parameter φ^+ is the objective value of the augmented Lagrangian.

$$\% \text{ Objective Value difference} = \frac{\varphi^+ - \varphi_x}{\varphi_x} * 100 \quad (4.16)$$

4.4.3.1. RTS-96 System

The objective value of the RTS-96 System is show in Table 4.1. In this table the PBGS results for various penalty factor ρ are compared with objective values obtained using EF method in Chapter 3. The simulation was carried out for four different values of penalty factors. In general, higher the penalty factor is larger the objective difference with EF solution.

Table 4.1 Comparison of PBGS with EF - set RTS96-10-S1

$\gamma=1.0, \beta=1.1$			EF = \$333,733	EF = 14 min
$\underline{\rho}, \bar{\rho}$	PBGS operating cost	PBGS solution time in minutes	Objective diff. w.r.t. EF	Time diff. w.r.t. EF
100	\$333,990	38	0.08%	171%
500	\$334,313	25	0.17%	79%
1,000	\$334,362	17	0.19%	21%
5,000	\$337,156	19	1.03%	36%

As shown in the above table, the PBGS objective solution is very close to the EF objective value, conveying that the PBGS solution will be good. This will be further confirmed in the application of PBGS on the two other systems. However, the time taken by the PBGS is much longer than EF.

4.4.3.2. The IEEE 118-bus System

The operating cost by PBGS under different values of the penalty factor ρ , and that of EF for IEEE 118-bus System, set IEEE118-10-S1 are listed in Table 4.2. Also shown in the table is the time taken to obtain PBGS solution and the comparison with the time taken to obtain EF solution.

Table 4.2 Comparison of PBGS with EF - set IEEE118-10-S1

$\gamma=1.0, \beta=1.1$			EF = \$851,152	EF = 15 min
$\underline{\rho}, \bar{\rho}$	PBGS operating cost	PBGS solution time in minutes	Objective diff. w.r.t. EF	Time diff. w.r.t. EF
500	\$851,149	149	0.00%	893%
1,000	\$851,097	106	-0.01%	607%
5,000	\$851,366	53	0.03%	253%
10,000	\$854,284	49	0.37%	227%

The PBGS solution under all ρ values, except for $\rho=10000$, is very close to the EF solution with the gap calculated to be less than 0.05%. When $\rho=1000$, the PBGS difference with respect to EF is -0.01%, signifying that its operating cost is slightly less (better) than EF. The negative gap can be attributed to the MIP optimality gap of 0.1% that was set in CPLEX. For all ρ , the PBGS gap with respect to EF is less than 0.4%. The EF solution, based on the state-of-the-art commercial MIP solver, is one of the most successful approaches in solving small- to medium-scale S-NCUC problems. Here, the EF solution is used to benchmark the PBGS solution. It is observed that the difference in the operating cost between the PBGS method and EF is very small for the IEEE-118 bus system. The results demonstrate the effectiveness of the PBGS solution on the medium-scale S-NCUC problem.

However, the time taken by the PBGS to obtain solution is far longer than the time taken by the EF. This is shown in the last column of the Table 4.2. This indicates that the PBGS solution does not have any advantage over the EF solution for small number of scenarios. Table 4.3 shows results for the same system with 50 scenarios. The PBGS objective values are very close to the EF solution (all are less than 1%). The time taken to obtain the PBGS solution is getting closer to the EF solution time compared with Table 4.1. This indicates that as the scenario increases time taken by the EF is increasing as mentioned earlier.

Table 4.3 Comparison of PBGS with EF - set IEEE118-50-S0

$\gamma=1.0, \beta=1.1$			EF = \$886,970	EF = 304 min
$\underline{\rho}, \bar{\rho}$	PBGS operating cost	PBGS solution time in minutes	Objective diff. w.r.t. EF	Time diff. w.r.t. EF
5,000	\$895,015	703	0.91%	131%
10,000	\$889,651	593	0.30%	95%
25,000	\$889,371	493	0.27%	62%
50,000	\$888,161	539	0.13%	77%

In the next section of this chapter, the modification to the PBGS will be discussed to improve computational efficiency.

4.4.3.3. The ERCOT-Like System

Three simulations are carried out with different penalty factors ρ for the ERCOT-like System ERCOT-30-S1 Table 4.4 shows the PBGS operating cost along with the time taken to obtain the solution. It is important to note that the EF is not implemented in the ERCOT-like System due to the enormous computational burden of EF (computationally infeasible). Therefore, no comparison can be made in this section to evaluate the PBGS solution for ERCOT-30-S1. The result for the ERCOT-like System will be evaluated in Chapter 5.

Table 4.4 PBGS solution for the ERCOT-Like System, set ERCOT-30-S1

$\gamma=1.0, \beta=1.1$		PBGS solution time in sequence (min)	Estimated PBGS solution time in parallel (min)
$\underline{\rho}, \bar{\rho}$	Operating cost by PBGS		
5,000	\$35,444,677	5955	267
10,000	\$35,480,631	5584	244
50,000	\$35,880,952	3251	137

It is fascinating to discuss the computational performance of PBGS as this is a common concern in large-scale applications. In Table 4.4, the column labeled “PBGS solution time in sequence” shows the solution time of S-NCUC on the ERCOT-like System with 30 scenarios,

where each scenario was solved sequentially without parallelization. Suppose one were to implement the proposed algorithm in the day-ahead operation. In that case, the time taken for sequentially solving each scenario of the ERCOT-like System is far from practical. The solution time can be drastically shortened using parallel computing, as shown in the column labeled “Estimated PBGS solution time in parallel” in Table 4.4. The time estimation is based on the longest time taken by a scenario in each iteration plus the time between MATLAB and GAMS interaction through the file I/O activities. The FESTIV environment used is in an academic setting and currently does not support the parallel execution. It is understood that the parallel computing environment is not a standard in ISOs or utilities, which may need hardware upgrades and adjustments to the existing day-ahead market timeline if the stochastic method is implemented. PBGS can still be used for short-term such as week-ahead stochastic planning studies and in day-ahead S-NCUC for systems smaller than ERCOT. The Fast PBGS algorithm discussed in the next section shortens the time considerably, making it attractive, especially for small and medium-sized systems.

4.5. Fast Penalty-Based Gauss-Seidel (Fast PBGS)

During the analysis of PBGS results for large-scale S-NCUC, it was observed that the computational efficiency of the PBGS could be improved by skipping scenarios that meet specific conditions from solving every iteration. This change resulted in an improved PBGS algorithm called "Fast PBGS" [104].

The key idea in Fast PBGS is explained using mathematical proposition followed by the proof. This idea is possible because both \mathbf{I} and \mathbf{Z} always being binary values. This would not be possible in other scenario-based algorithms where calculation of \mathbf{Z} is calculated as real numbers. For example, in PHA the calculation of \mathbf{Z} at an iteration k is accomplished as probability (\mathbf{Pr}) weighted average of scenario decisions \mathbf{I} as shown in (1). Also, \mathbf{I} is solved as integer value (MIP).

$$\mathbf{Z}_k := \sum_{s \in S} Pr^s \cdot \mathbf{I}_k^s \quad (4.17)$$

This calculation is compared to the one done in PBGS in (4.8), which is shown below in (4.18). The implementable \mathbf{Z} is a binary value whose range is the same as that of \mathbf{I} making the comparison possible.

$$Z_{i,t} := \begin{cases} 1, & \text{if } \sum_{s \in S} (1 - I_{i,t}^s) \cdot \underline{\rho}_{i,t}^s < \sum_{s \in S} I_{i,t}^s \cdot \bar{\rho}_{i,t}^s \\ 0, & \text{if } \sum_{s \in S} (1 - I_{i,t}^s) \cdot \underline{\rho}_{i,t}^s > \sum_{s \in S} I_{i,t}^s \cdot \bar{\rho}_{i,t}^s \\ 0 \text{ or } 1, & \text{otherwise} \end{cases} \quad (4.18)$$

Corollary 1: For scenario s in PBGS, $(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*})$ is also the optimal solution at the $k+1$ th iteration if the following two conditions are satisfied at the k th iteration:

- (1) The optimal commitment decisions obtained are the same as the pre-update implementable, i.e.,

$$\mathbf{I}_k^{S*} = \mathbf{Z}_{k-1} \quad (4.19)$$

- (2) The post-update implementable remains unchanged, i.e.,

$$\mathbf{Z}_k = \mathbf{Z}_{k-1} \quad (4.20)$$

Proof 2: The objective function of the PBGS in an abbreviated form is given as follows, where all the constraints (3.2) – (3.7) are applicable.

$$\varphi^+(\mathbf{I}, \mathbf{p}, \boldsymbol{\rho}) = C(\mathbf{I}, \mathbf{p}) + \boldsymbol{\rho} \psi(\mathbf{I} - \mathbf{Z}) \quad (4.21)$$

where $C(\mathbf{I}, \mathbf{p})$ is a scenario-wise objective function and $\boldsymbol{\rho} \psi(\mathbf{I} - \mathbf{Z})$ is the augmented Lagrangian term.

Proof by contradiction is used here. Let $(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*})$ denote the optimal solution obtained in scenario s at the k^{th} iteration of PBGS, i.e.,

$$(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*}) = \underset{\mathbf{I}, \mathbf{p}}{\operatorname{argmin}} [C(\mathbf{I}_k^s, \mathbf{p}_k^s) + \boldsymbol{\rho}_k \cdot \psi(\mathbf{I}_k^s - \mathbf{Z}_{k-1})] \quad (4.22)$$

Similarly, let $(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*})$ be the optimal solution in scenario s at the $k+1^{th}$ iteration of the PBGS. It is to be noted that $\boldsymbol{\rho}_{k+1} \geq \boldsymbol{\rho}_k \geq 0$ in PBGS. It is apparent that the optimal solution obtained in k^{th} iteration is a feasible solution to the $k+1^{th}$ iteration since the solution is from the feasible region Λ^s defined in (3.8). Feasible values $(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*})$ is defined as follows:

$$(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}) = \underset{\mathbf{I}, \mathbf{p}}{argmin} [C(\mathbf{I}_{k+1}^S, \mathbf{p}_{k+1}^S) + \boldsymbol{\rho}_{k+1} \cdot \psi(\mathbf{I}_{k+1}^S - \mathbf{Z}_k)] \quad (4.23)$$

Assume $(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*})$ in (4.22) is not equal to $(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*})$ in (4.23). Under this assumption, in view of its feasibility, we have

$$\varphi^+(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}, \boldsymbol{\rho}_{k+1}) < \varphi^+(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*}, \boldsymbol{\rho}_k) \quad (4.24)$$

Substituting (4.21) into (4.24), we get

$$C(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}) + \boldsymbol{\rho}_{k+1} \cdot \psi(\mathbf{I}_{k+1}^{S*} - \mathbf{Z}_k) < C(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*}) + \boldsymbol{\rho}_k \cdot \psi(\mathbf{I}_k^{S*} - \mathbf{Z}_k) \quad (4.25)$$

Applying conditions (4.19) and (4.20) in Corollary 1 to (4.25), we get:

$$C(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}) + \boldsymbol{\rho}_{k+1} \cdot \psi(\mathbf{I}_{k+1}^{S*} - \mathbf{Z}_{k-1}) < C(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*}) \quad (4.26)$$

For any given solution $(\tilde{\mathbf{I}}_k^s, \tilde{\mathbf{p}}_k^s)$, a higher $\boldsymbol{\rho}$ leads to a higher objective function value in terms of a higher penalty term due to the structure of φ^+ . Hence,

$$\begin{aligned} C(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}) + \boldsymbol{\rho}_k \cdot \psi(\mathbf{I}_{k+1}^{S*} - \mathbf{Z}_{k-1}) \\ \leq C(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}) + \boldsymbol{\rho}_{k+1} \cdot \psi(\mathbf{I}_{k+1}^{S*} - \mathbf{Z}_{k-1}) \end{aligned} \quad (4.27)$$

Combining (4.26) and (4.27) results in the following condition

$$C(\mathbf{I}_{k+1}^{S*}, \mathbf{p}_{k+1}^{S*}) + \boldsymbol{\rho}_k \cdot \psi(\mathbf{I}_{k+1}^{S*} - \mathbf{Z}_{k-1}) < C(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*}) \quad (4.28)$$

The left-hand side of (4.28) is exactly the objective function (4.21) in the k^{th} iteration of PBGS. Its value cannot be better than that of the optimal solution $(\mathbf{I}_k^{S*}, \mathbf{p}_k^{S*})$, which is written as:

$$\mathcal{C}(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*}) + \rho_k \cdot \psi(\mathbf{I}_k^{S^*} - \mathbf{Z}_{k-1}) \leq \mathcal{C}(\mathbf{I}_{k+1}^{S^*}, \mathbf{p}_{k+1}^{S^*}) + \rho_k \cdot \psi(\mathbf{I}_{k+1}^{S^*} - \mathbf{Z}_{k-1}) \quad (4.29)$$

According to (4.19) in Corollary 1, we have

$$\mathcal{C}(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*}) = \mathcal{C}(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*}) + \rho_k \cdot \psi(\mathbf{I}_k^{S^*} - \mathbf{Z}_{k-1}) \quad (4.30)$$

Combining (4.28) – (4.30), we get

$$\begin{aligned} \mathcal{C}(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*}) &= \mathcal{C}(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*}) + \rho_k \cdot \psi(\mathbf{I}_k^{S^*} - \mathbf{Z}_{k-1}) \\ &\leq \mathcal{C}(\mathbf{I}_{k+1}^{S^*}, \mathbf{p}_{k+1}^{S^*}) + \rho_k \cdot \psi(\mathbf{I}_{k+1}^{S^*} - \mathbf{Z}_{k-1}) \\ &\leq \mathcal{C}(\mathbf{I}_{k+1}^{S^*}, \mathbf{p}_{k+1}^{S^*}) + \rho_{k+1} \cdot \psi(\mathbf{I}_{k+1}^{S^*} - \mathbf{Z}_{k-1}) \\ &< \mathcal{C}(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*}) \end{aligned} \quad (4.31)$$

Apparently, (4.31) does not hold. This indicates the assumption that $(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*})$ made in (4.22) is not equal to $(\mathbf{I}_{k+1}^{S^*}, \mathbf{p}_{k+1}^{S^*})$ in (4.23) cannot hold. Therefore, $(\mathbf{I}_{k+1}^{S^*}, \mathbf{p}_{k+1}^{S^*})$ must be equal to $(\mathbf{I}_k^{S^*}, \mathbf{p}_k^{S^*})$.

■

Proposition 1: Scenario s does not need be solved as long as equations (4.19) and (4.20) hold in successive iterations.

At iteration k , let us assume that both conditions 1 and 2 of the Corollary are met by scenario s . Let us also assume that Z has not changed in the $k + 1^{th}$ iteration, following Proof 2, and solving scenario s in the $k + 1^{th}$ iteration can be skipped. If Z continues to remain the same in subsequent iterations, one can prove (by induction), using Proof 2 that scenario s can be skipped in the subsequent iterations as well. ■

Based on the above discussion, the proposed Fast PBGS algorithm is shown below in Algorithm 2, in which indices i and t are omitted for brevity.

4.5.1. Fast PBGS Algorithm

Algorithm 2: Fast Penalty-Based Gauss-Seidel Algorithm

```

1 Initialization  $\rho, \gamma, \beta, \epsilon, l_{max}, \omega$ 
2 foreach  $s \in S$  do
3    $(I^s, p^s) \leftarrow \underset{I^s, p^s}{\operatorname{argmin}} \{ \varphi^+(I^s, p^s, \omega^s, \rho^s = 0) \mid (I^s, p^s) \in \Lambda^s \}$ 
4 end
5 Calculate  $Z^0 \leftarrow \operatorname{round}(\sum_{s \in S} P r^s \cdot I^s)$ 
6 while  $(\sum_{s \in S} \|I^s - Z\|_2^2 \neq 0)$  do
7   for  $l = 1, \dots, l_{max}$  do
8     foreach  $s \in S$  do
9       if  $((I^s \neq Z^{l-1}) \text{ or } (Z^{l-1} \neq Z^l))$  then
10         $(I^s, p^s, \phi^{s,l}) \leftarrow \underset{I^s, p^s}{\operatorname{argmin}} \{ \varphi^+(I^s, p^s, \omega^s, \rho^s) \mid (I^s, p^s) \in \Lambda^s \}$ 
11      end
12    end
13    Calculate  $Z^l$  using (4.8)
14     $\Gamma \leftarrow \phi^l - \phi^{l-1}$ 
15    if  $(|\Gamma| \leq \epsilon) \text{ or } (l = l_{max})$  then
16       $Z \leftarrow Z^l$ 
17      break
18    end
19     $l \leftarrow l + 1$ 
20  end
21  Update  $\underline{\rho}^s$  and  $\overline{\rho}^s$  using (4.9)
22  Update  $\alpha$  using (4.12)
23 end
24 return  $Z$ 

```

4.5.2. Initialization of Implementable \mathbf{Z}

The implementable \mathbf{Z} should be initialized for all the scenarios. There are many ways to initialize \mathbf{Z} , and the PBGS algorithm developed in [101] discusses the first two methods. The third method proposed here is used in this research yielding faster solution.

Method 1: This method is straightforward and does not require any additional efforts. The disadvantage of the method is longer solution time.

$$\mathbf{Z} := \mathbf{0} \quad (4.32)$$

Method 2: This method is shown in Algorithm 2, lines 2-5. After solving all the scenarios without augmenting function, the rounded-up sum of the weighted average of the individual scenario status for the given unit at a given interval is assigned to \mathbf{Z} . The method requires an additional step of running all the scenarios in the beginning. This may seem time consuming but it helps improve overall computation time.

$$Z_{i,t} := \left\lceil \sum_{s \in S} Pr^s \cdot I_{i,t}^s \right\rceil, \quad \forall t \in NT, \forall i \in NI \quad (4.33)$$

Method 3: This is the method devised and used in the simulations conducted as part of this research. The initial implementable can be set to one of the scenarios' commitment decisions from the set of scenarios that are being evaluated. The scenario chosen to be the initial implementable is a scenario with the maximum number of online statuses of all the units among all the scenarios. If more than one scenario has the same number of the online status of all the units, choose the scenario with the lowest production cost.

$$\mathbf{Z} := \{\mathbf{x} \in X \mid \forall \mathbf{y} \in X, f(\mathbf{x}) > f(\mathbf{y})\} \quad (4.34)$$

where $X = \{\mathbf{I} \mid s \in S\}$

$f: S \rightarrow \mathbb{R}$ defined as:

$$f(s) := \sum_{i=1}^{NI} \sum_{t=1}^{NT} I_{i,t}^s$$

A comparison of the three methods are presented in Chapter 6 to further explain the importance of the initialization.

4.5.3. Comparison of PBGS vs Fast PBGS

This section compares the results obtained through PBGS and Fast PBGS using Algorithm 1 and Algorithm 2, respectively. Through proof two, it is established that the objective value obtained using Fast PBGS is the same as PBGS. Therefore, only the computation time taken to obtain the solution is compared.

Plots showing the number of scenarios solved as the iteration progresses towards convergence are presented in this section to demonstrate how Fast PBGS saves computational time. It should be noted that in the PBGS algorithm, all scenarios are solved in every iteration. Overall, one can observe that after a few initial iterations, the number of scenarios solved is decreasing as the iteration progresses.

4.5.3.1. RTS-96 System

The scenario set RTS96-10-S1, discussed in section 4.4.3.1, is used in the comparison here. The comparison is shown in Table 4.5. While the final result and number of iterations to reach the convergence are the same for both PBGS and the Fast PBGS, the time taken to achieve the result for the latter is an average of 30% faster.

Table 4.5 Comparison of PBGS with Fast PBGS - set RTS96-10-S1

ρ	Time in min		% time gain
	PBGS	Fast PBGS	
500	38	27	28%
1,000	25	16	35%
5,000	17	12	27%
10,000	19	13	31%

Figure 4.4 shows the number of scenarios solved in each iteration for the RTS-96 System for various penalty factor. For the penalty factor 10000, in the first seven iterations all 10 scenarios are solved. In the next four iterations, 2 to 3 scenarios agree with the implementable and therefore these scenarios are skipped from solving. In the 12th iteration, all 10 scenarios are solved as the implementable changes. Starting at the 15th iteration through the 27th, most of the scenarios are not solved. In the 28th iteration, 9 scenarios are solved. The implementable agrees with one scenario holding out from the 24th iteration and giving up in the 28th while forcing 9 scenarios agreeing with the implementable before. Figure 4.4 also shows the Fast PBGS results for other penalty factors. As was discussed in earlier, the higher the penalty factor the faster is the reduction in the number of scenarios solved per iteration.

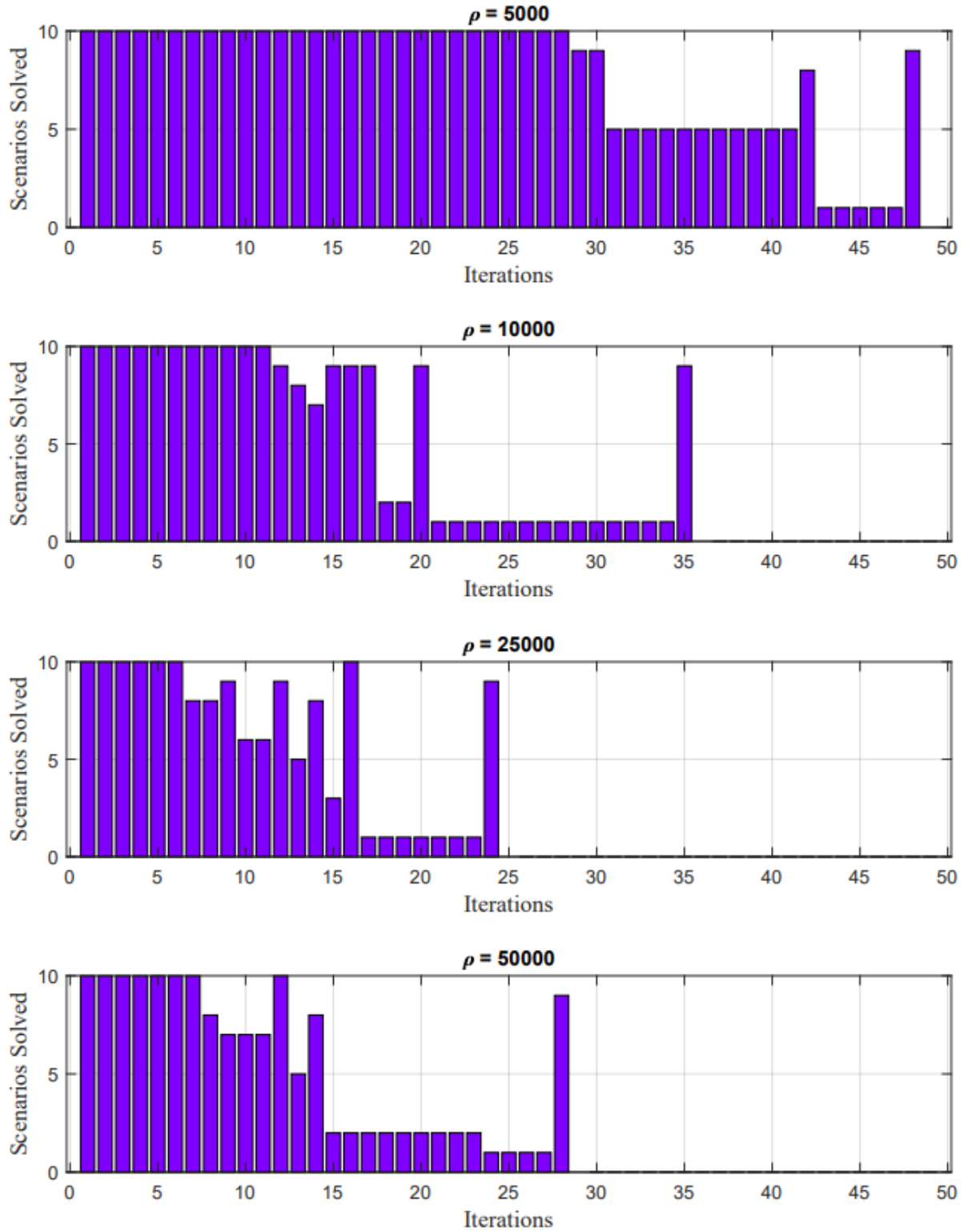


Figure 4.4 Number of scenarios solved in each iteration for the RTS-96 System using Fast PBGS

4.5.3.2. The IEEE 118-bus System

Like in section 4.4.3.2, the computational time between PBGS and Fast PBGS for both ten scenarios and 50 scenario sets is used. Table 4.6 shows the results for the IEEE 118-bus System, set IEEE118-10-S1. For this set, the time taken for Fast PBGS to obtain a solution is an average of about 19% less than PBGS for the IEEE 118-bus System.

Table 4.6 Comparison of PBGS with Fast PBGS - set IEEE118-10-S1

ρ	Time in min		% time gain
	PBGS	Fast PBGS	
500	149	120	19%
1,000	106	89	16%
5,000	53	45	15%
10,000	49	37	25%

Computation comparison for set IEEE118-50-S0 is shown in Table 4.7. The computational time saved by Fast PBGS over PBGS is an average of 50%. The computational time saving is about 30% more than for the ten-scenario set. It appears that the more the scenarios, the more time saved.

Table 4.7 Comparison of PBGS with Fast PBGS - set IEEE118-50-S0

ρ	Time in min		% time gain
	PBGS	Fast PBGS	
5,000	703	299	57%
10,000	593	282	52%
25,000	493	291	41%
50,000	539	271	50%

In Figure 4.5, for penalty factor 500 of the IEEE 118-bus, all scenarios are solved for the first 23 iterations, then the number of scenarios solved starts decreasing. Starting at the 39th iteration, only one iteration is solved except at the 43rd iteration where all ten scenarios are solved. Note that in the last iteration only one scenario is not solved. This tells that the implementable which was agreeing with nine scenarios has changed to agree with one which was not forcing all nine previously agreed scenarios to solve earlier.

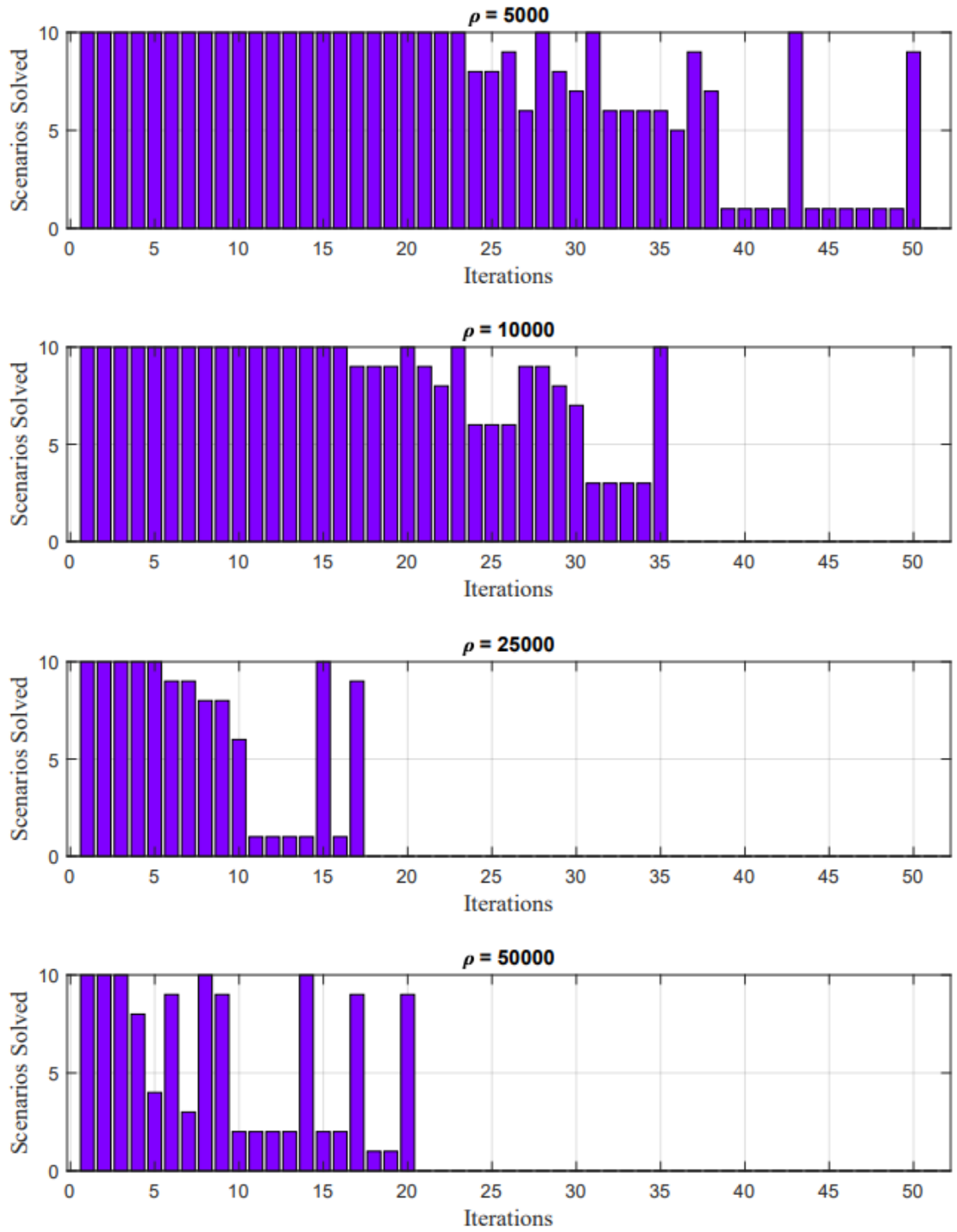


Figure 4.5 Number of scenarios solved in each iteration for the IEEE 118-bus System using Fast PBGS

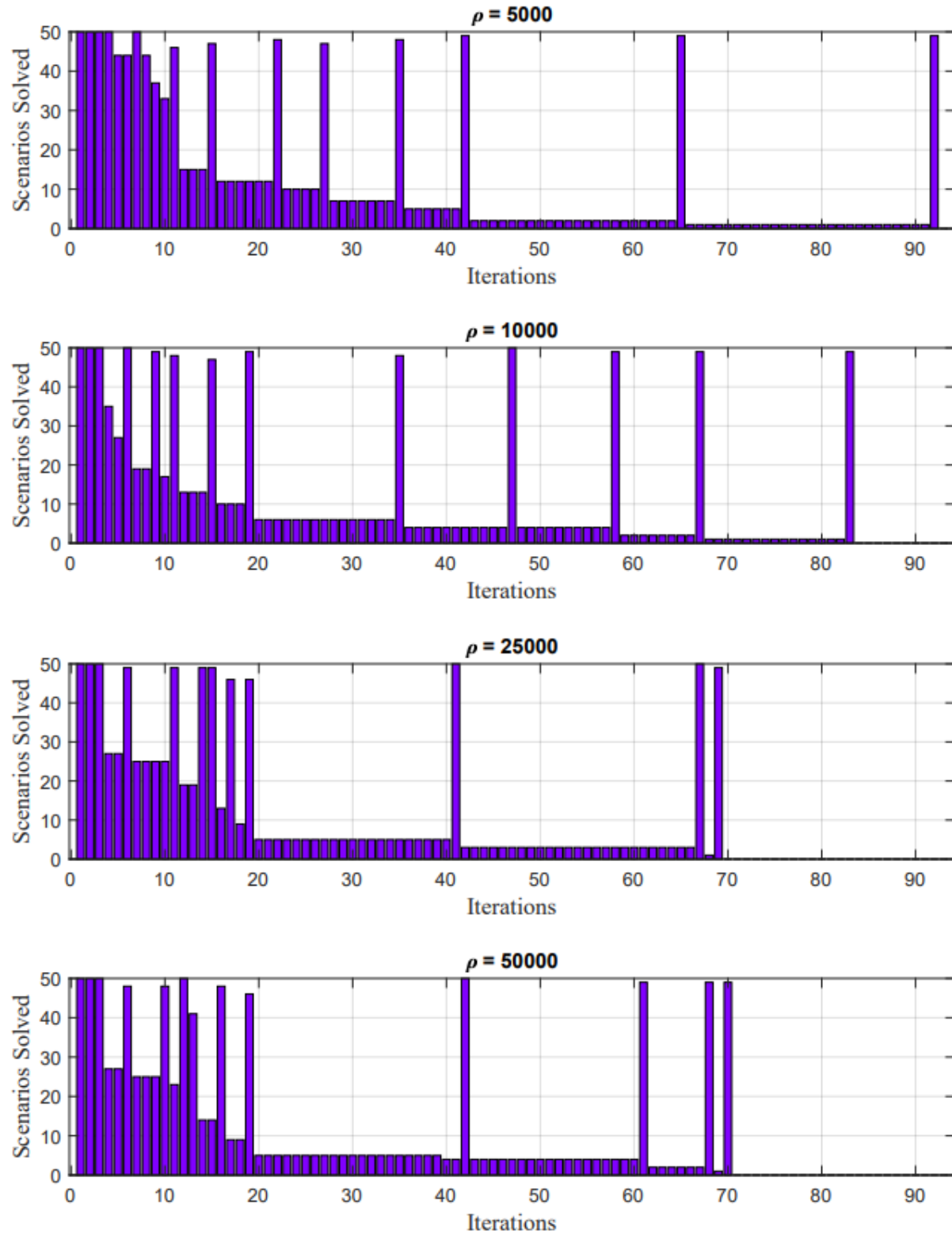


Figure 4.6 Number of scenarios solved in each iteration for the IEEE 118-bus System with 50 Scenarios

Figure 4.6 shows the actual number of scenarios solved at each iteration for the IEEE 118-bus System with 50 scenarios. With only a few first iterations, savings in time are observed, especially for the higher penalty factors.

4.5.3.3. ERCOT-like Large System

Finally, Table 4.8 shows the results for the sizeable ERCOT-like System. The set, for this large system, the computational time saved by Fast PBGS over the PBGS is from 34% to 39%, with an average of around 36%. In section 4.4.3.3, there was a discussion on the practical implementation of PBGS in the real world for large systems. The research made significant improvements in reducing the computational time through Fast PBGS. Even if it is not practical for the real-world application due to market timeline constraints, Fast PBGS is attractive to evaluate the benefits of S-NCUC over NCUC.

Table 4.8 Comparison of PBGS with the Fast PBGS - set ERCOT-30-S1

ρ	Time in min		% time gain
	PBGS	Fast PBGS	
5,000	5955	3921	34%
10,000	5584	3444	38%
50,000	3251	1981	39%

In Figure 4.6, the number of scenarios solved over iterations are shown for the ERCOT-like large system. This system uses 30 scenarios. Compared to the other two systems, significant time is saved here, which could be attributed to the larger number of scenarios than that has been used in the other two systems.

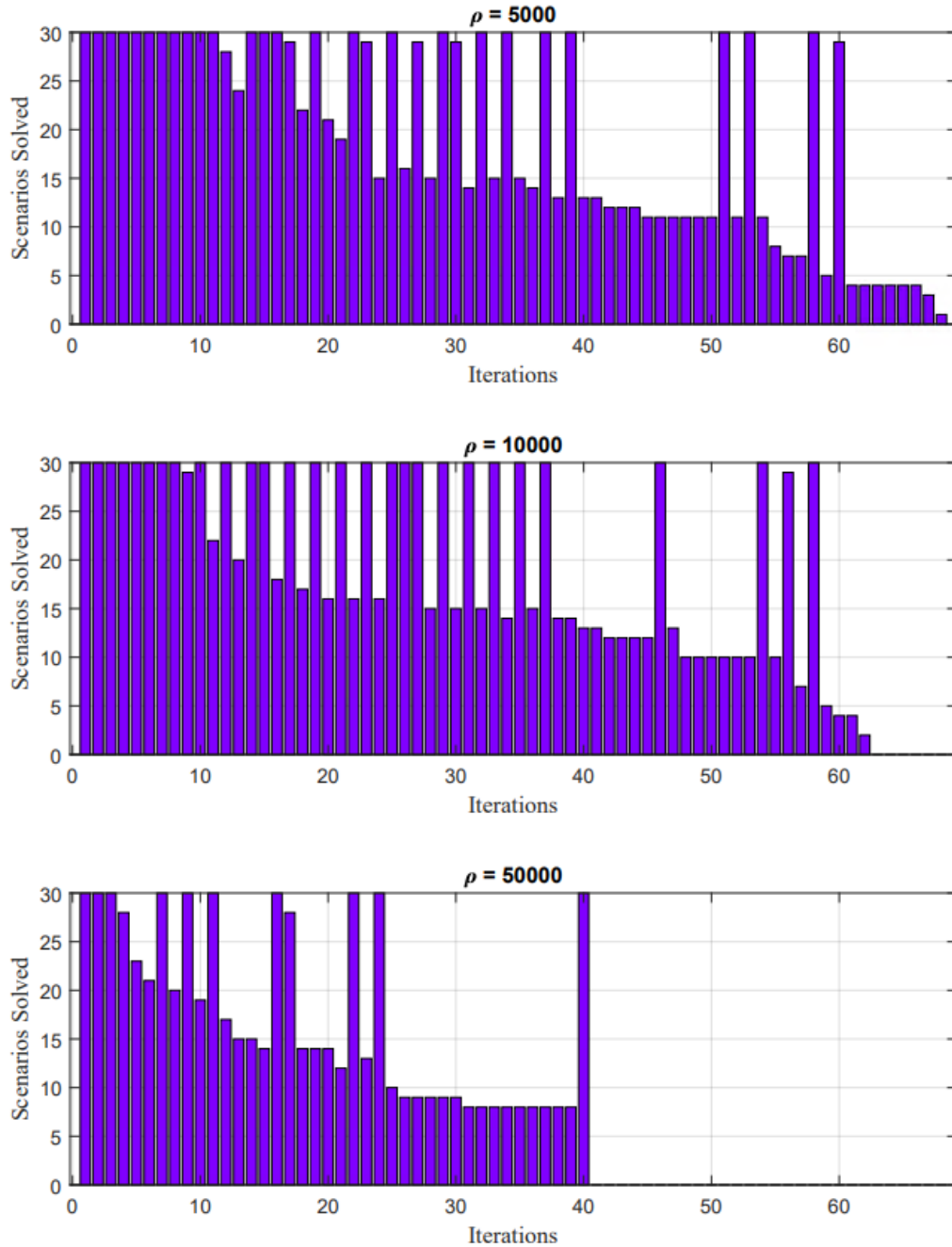


Figure 4.7 Number of scenarios solved in each iteration for the ERCOT-like large System using Fast PBGS

Comparing Figures 4.6 and 4.7 with Figures 4.4 and 4.5 (a smaller number of scenarios), one can observe that the higher the number of scenarios, the more time saved. It is evident from the above figures how the Fast PBGS saves time. This saving in computational time is a significant improvement over PBGS.

4.6. Summary

This chapter started with a discussion on the decomposition of the S-NCUC problem is discussed. The application of augmented Lagrangian in decomposing and solving individual problems is presented. In doing so, the difference between proximal Lagrangian and sharp Lagrangian was explained. It has been shown that sharp Lagrangian could reach the ‘dents’ where proximal Lagrangian is unable to do so. Reaching the ‘dent’ could achieve the zero-duality gap. The PBGS is applied to solve the S-NCUC problem through a detailed discussion of the different components of the algorithm. The discussion emphasized obtaining a primal solution that eliminates other processes that would have been needed to recover an implementable solution. A proof of convergence of the PBGS algorithm was given as well. Results for all three systems were presented. The system that was not solvable using EF in Chapter 3 was solved using the decomposition technique. However, the solution obtained for the ERCOT-like Large System is yet to be evaluated and will be discussed in the next chapter. To gain the computational efficiency of the PBGS algorithm, certain scenarios that meet specific criteria are not solved in any given iteration. Mathematical proof of this improvement of PBGS, which is called Fast PBGS, is provided. This chapter compared Fast PBGS solution with PBGS solution showing average time saving of 35% for ERCOT-like Large System and 50% for IEEE 118-bus System with 50 scenarios demonstrating advantages of the former.

5. Assessing S-NCUC Solution Quality

As discussed in detail in the previous chapter, the PBGS algorithm used in solving S-NCUC employs exact augmented Lagrangian. Since the S-NCUC is a non-convex problem due to binary values in the objective function, the solution obtained often is a local minimum. There could be more than one such local minima, and it is essential to evaluate the quality of the solution obtained using one of the lower-bound methods. Furthermore, it is vital to keep the computational cost of the lower-bound methods to a minimum while still thriving for the best lower-bound one can get. The Fast PBGS solution will be compared with one of the widely used scenario-wise decomposition methods, the PHA. This chapter includes a section for Out-of-Sample testing and parameter discussion each.

5.1. Lower Bound and Other Methods

Unlike other solution methods such as Benders decomposition method, the PBGS method does not rely on the bounds on the optimal production cost to form termination criteria. This lack of bounds makes us look into a method that needs to be solved separately in order to obtain a lower bound to assess the PBGS solution quality. Both lower-bound methods and primal methods are discussed here. The primal method is used for benchmarking the obtained PBGS solutions.

5.2. Extensive Formulation (EF)

This is a primal method discussed in detail in Chapter 3, the deterministic equivalent of the stochastic formulation. Extensive Formulation (EF) solves the stochastic problem with scenarios as one big single deterministic problem as formulated in (3.1) with constraints (3.2) – (3.7). Of course, if one can solve EF, there is no need for other algorithms such as PHA and PBGS. Since the computational time for EF formulation grows exponentially as the problem size increases, current off-the-shelf optimization applications cannot solve the S-NCUC, requiring decomposition techniques. As mentioned in Chapter 3, the large ERCOT-like System was not solvable by EF algorithm.

5.3. Relaxed Extensive Formulation (Re-EF)

In this method the integrality constraint in S-NCUC is relaxed and solved to obtain lower bound. This is the same EF formulation except that linear programming (LP) is applied. The only difference is in (3.1)

$$\mathbf{I} \in \{0,1\}^{NI \times NT \times S} \text{ will relax to } \mathbf{I} \in \{\mathbb{R}\}^{NI \times NT \times S}$$

This Re-EF could work for an extensive system where EF formulation might fail. However, the lower bound obtained is not as good as, the lower bound obtained from other methods. This is expected behavior for such formulations that relaxes integrality constraints. This research formulated Re-EF for one system where simulation has been conducted. In Table 5.1, the difference calculated using Re-EF is shown. As can be seen, the difference is above 10%. This poor gap is not due to the sub-optimality of the PBGS. Instead, due to the poor lower bound of Re-EF. This is evident from the fact that the difference calculated using EF is very little.

Table 5.1 Comparison of Fast PBGS with Re-EF

$\gamma=1.0, \beta=1.1$		EF = \$333,733 EF Relaxed = \$298,994	
$\underline{\rho}, \bar{\rho}$	Operating cost by Fast PBGS	Diff. w.r.t. EF	Diff. w.r.t. EF-Relaxed
100	\$333,990	0.08%	11.70%
500	\$334,313	0.17%	11.81%
1,000	\$334,362	0.19%	11.83%
5,000	\$337,156	1.03%	12.76%

5.4. Progressive Hedging Algorithm (PHA)

The PHA algorithm was briefly discussed in Chapter 2. First proposed by Rockafeller and Wets [51] to solve stochastic optimization problems involving continuous variables has been applied with limited success in problems involving integer variables. Gade *et.al.* have used this algorithm to obtain lower bounds in [55]. PHA is similar to PBGS except for the augmenting

function used in (4.2). PHA uses square of l_2 norm as augmenting function. With this augmenting function and setting $\alpha = 1$ in (4.2), the PHA objective function is given as:

$$\begin{aligned} \Theta^+(I, p, \omega, \rho) = \\ \min \sum_{s=1}^S Pr^s \left(\sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left(NL_i \cdot I_{i,t}^s + SU_{i,t} \cdot IU_{i,t}^s + \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) + \right. \right. \right. \\ \left. \left. \left. \omega_{i,t}^s \cdot (I_{i,t}^s - Z_{i,t}) + \frac{\rho}{2} \|I_{i,t}^s - Z_{i,t}\|_2^2 \right) \right. \right. \\ \left. \left. + (AL_t^s + LL_t^s) \cdot VOLL + \sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right] \right) \end{aligned} \quad (5.1)$$

As mentioned in Chapter 3, $\sum_{s \in S} Pr^s \omega^s = 0$ is enforced for the augmented Lagrangian function (4.1) to be bounded from below. The NAC defines a subspace \mathcal{N} and the optimality conditions require that the dual variable (ω) lie in the subspace of \mathcal{N}^\perp [51]. This requirement, $\sum_{s \in S} Pr^s \omega^s = 0$ is enforced by updating ω in every iteration of the PHA algorithm. This makes $\omega_{i,t}^s \cdot Z_{i,t}$ term in (5.1) vanish. Accounting this change, (5.1) is rewritten as (5.2).

$$\begin{aligned} \Theta^+(I, p, \omega, \rho) = \\ \min \sum_{s=1}^S Pr^s \left(\sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left((NL_i + \omega_{i,t}^s) \cdot I_{i,t}^s + SU_{i,t} \cdot IU_{i,t}^s + \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) \right. \right. \right. \\ \left. \left. \left. + \frac{\rho}{2} \|I_{i,t}^s - Z_{i,t}\|_2^2 \right) \right. \right. \\ \left. \left. + (AL_t^s + LL_t^s) \cdot VOLL + \sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right] \right) \end{aligned} \quad (5.2)$$

Writing (5.2) for each scenario:

$$\begin{aligned} \Theta^{+s}(\mathbf{I}^s, \mathbf{p}^s, \boldsymbol{\omega}^s, \boldsymbol{\rho}^s) = \\ \min \sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left((NL_i + \omega_{i,t}^s) \cdot I_{i,t}^s + SU_{i,t} \cdot IU_{i,t}^s + \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) \right. \right. \\ \left. \left. + \frac{\rho}{2} \|I_{i,t}^s - Z_{i,t}\|_2^2 \right) \right. \\ \left. + (AL_t^s + LL_t^s) \cdot VOLL + \sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right] \end{aligned} \quad (5.3)$$

The implementable is updated as follows

$$Z_{i,t} := \sum_{\forall s \in S} Pr^s \cdot I_{i,t}^s \quad \forall i \in NI, \forall t \in NT \quad (5.4)$$

The dual variable $\boldsymbol{\omega}$ is updated in the $k+1^{th}$ iteration as shown below.

$$\boldsymbol{\omega}_{i,t}^{s,k+1} := \boldsymbol{\omega}_{i,t}^{s,k} + \boldsymbol{\rho}(I_{i,t}^{s,k} - \mathbf{Z}_{i,t}^k) \quad \forall i \in NI, \forall t \in NT, \forall s \in S \quad (5.5)$$

The termination criteria is defined as

$$\sqrt{\sum_{s \in S} p_s \|I_s^k - \mathbf{z}^k\|_2^2} < \epsilon, \text{ terminate} \quad (5.6)$$

To calculate the lower bound, (5.3) is used without augmenting Lagrangian term (just Lagrangian) as shown below

$$\Theta^s = \sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left((NL_i + \omega_{i,t}^s) \cdot I_{i,t}^s + SU_{i,t} \cdot IU_{i,t}^s + \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) \right) \right. \\ \left. + (AL_t^s + LL_t^s) \cdot VOLL + \sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right] \quad (5.7)$$

The above problem can be solved for each scenario and the total cost is computed as follows:

$$\phi = \min \sum_{s=1}^S Pr^s(\Theta^s) \quad (5.8)$$

The complete algorithm used in calculating the lower bound using PHA is given in Algorithm 3 below.

Note that the ϕ^k calculated at each iteration k in Step 14 of Algorithm 3 is a lower bound. So, one does not have to wait for convergence to occur in order to obtain the lower bound.

Algorithm 3: Progressive Hedging Algorithm - Includes Lower-Bound Computation

```
1 foreach  $s \in S$  do
2    $(I^s, p^s) \leftarrow \underset{I_k^s, p_k^s}{\operatorname{argmin}} \{ \Theta^+(I^s, p^s, \omega^s = 0, \rho^s = 0) \mid (I^s, p^s) \in \Lambda^s \}$ 
3 end
4  $Z^0 \leftarrow \sum_{s \in S} Pr^s \cdot I^s$ 
5 foreach  $s \in S$  do
6    $\omega^{s,0} \leftarrow \rho \cdot (I^s - Z^0)$ 
7 end
8 while  $(\sum_{s \in S} \|I^s - Z\|_2^2 > \epsilon)$  do
9   for  $l = 1, \dots, l_{max}$  do
10    foreach  $s \in S$  do
11       $(I^s, p^s) \leftarrow \underset{I_k^s, p_k^s}{\operatorname{argmin}} \{ \Theta^+(I^s, p^s, \omega^s, \rho^s) \mid (I^s, p^s) \in \Lambda^s \}$ 
12       $\phi^{s,l} \leftarrow \{ \Theta^s(I^s, p^s, \omega^s) \mid (I^s, p^s) \in \Lambda^s \}$ 
13    end
14     $Z^l \leftarrow \sum_{s \in S} Pr^s \cdot I^s$ 
15    foreach  $s \in S$  do
16       $\omega^{s,l} \leftarrow \omega^{s,l-1} + \rho \cdot (I^s - Z^l)$ 
17    end
18     $\phi^l \leftarrow \sum_{s \in S} Pr^s \cdot I^s \phi^{s,l}$ 
19     $l \leftarrow l + 1$ 
20  end
21 return  $\phi^l$ 
```

5.4.1. Comparison of Fast PBGS with PHA

Since the scope of this research is restricted to the scenario-based decomposition methods, the comparison is also confined to the methods within this approach. In the literature, there exists a stage-wise decomposition approach, such as Benders decomposition. However, the stage-wise decomposition approach may eventually grow into a computationally intractable problem as the iteration progresses for the size of the problem considered in this dissertation. Therefore, the cross-comparison between the scenario-based and stage-based decomposition methods is out of the scope of this research and can be systematically conducted in future work.

5.4.2. The IEEE 118-bus System

Figure 5.1 shows a comparison of the Fast PBGS with the PHA for the IEEE 118-bus System. The second y-axis is the percentage difference in the objective value per (5.1). A negative percentage difference indicates that the F-PBGS leads to a better solution in terms of the objective value, and vice versa. Table 5.2 shows the data values of Figure 5.1.

The comparison between the Fast PBGS and PHA is carried out with five different ρ values ranging from 500 to 20,000. The Fast PBGS has one additional tuning parameter β that is used in the calculation of convergence accelerator α in (4.12). For each of these five ρ values, studies used two β values, i.e., 1.11 and 1.25. It is seen in Figure 6.1 when β is 1.25, the Fast PBGS solves faster than it does when β is set to 1.11, but with a slightly higher objective value. The difference in the solution time between the Fast PBGS and the PHA is much more apparent with a lower ρ . Overall, the Fast PBGS yields a much faster solution than the PHA. By using the Fast PBGS, an average of 42% and 61% saving in solution time with β values equal to 1.11 and 1.25, respectively, was recorded.

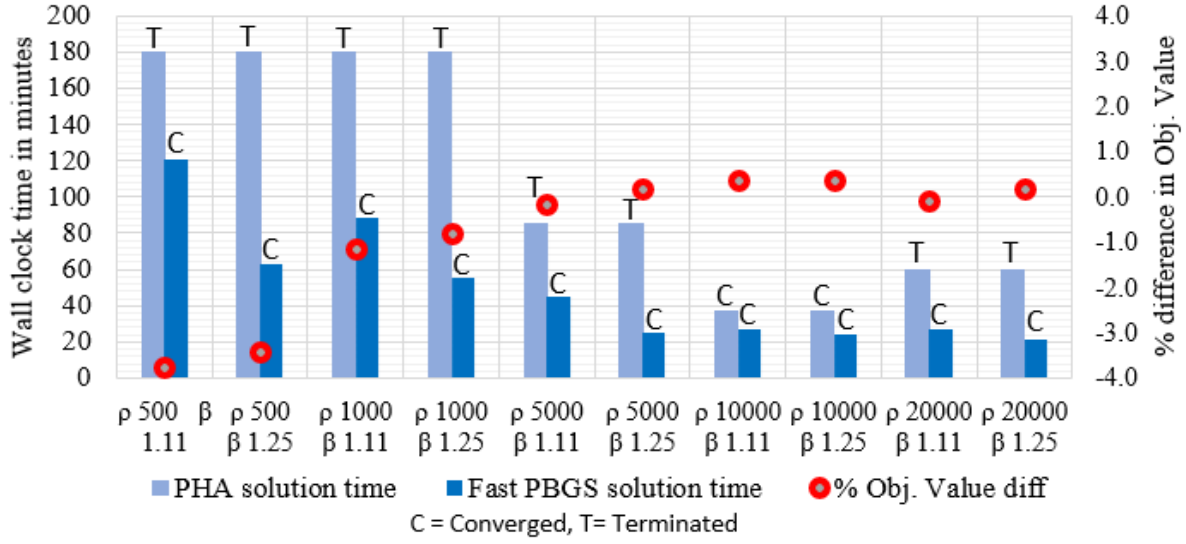


Figure 5.1 Comparison of Fast PBGS with PHA for the IEEE 118-bus System

In Figure 5.1, all Fast PBGS simulations converge with zero violations. In contrast, none of the PHA solutions converges except for the case when $\rho = 10000$. The convergence tolerance ϵ for PHA is set to 0.01. At a lower ρ , a solution time limit of 180 minutes was set. At a higher ρ (i.e., 5000 and 20000), the PHA algorithm is terminated if no progress in violation is made for five consecutive iterations. Additionally, a terminated PHA algorithm requires rounding-off the fractional implementable values obtained, followed by Unit Commitment to ensure its primal feasibility.

Table 5.2 Comparison of Fast PBGS with PHA for the IEEE 118-bus System

		EF Obj Value = \$851,152			EF Time in min = 15		
		ρ	500	1000	5000	10000	20000
Objective Value	PHA	\$884,333	\$861,043	\$852,711	\$851,259	\$854,895	
	F-PBGS	\$851,149	\$851,097	\$851,366	\$854,284	\$854,284	
Time in min	PHA	180	180	85	37	60	
	F-PBGS	120	89	45	27	27	

In Figure 5.1, for a comparatively small ρ , i.e., when $\rho=500$ and 1000, the Fast PBGS outperforms the PHA by 2% to 4% in terms of the objective value, respectively. This difference is

attributed to the rounding of the PHA implementable values, resulting in overcommitment of generators. When ρ is increased, the difference in the objective values decreases. When $\rho=10000$, the PHA converges and no rounding is required. The PHA performs slightly, i.e., around 0.5%, better than the Fast PBGS. The above results in Figure 5.1 suggest that the significant improvement in computational efficiency makes the Fast PBGS a better choice than the PHA on the medium-scale system.

5.4.3. The ERCOT-Like System

For the ERCOT-like large-scale system, the research set $\beta = 1.11$ and focus exclusively on the sensitivity studies with respect to ρ values. Comparative results between the Fast PBGS and the PHA are shown in Figure 5.2. It is seen that the Fast PBGS outperformed the PHA in terms of the solution time, whereas the objective value of PHA is slightly better than that of the Fast PBGS in all three cases. The average execution time saved by the Fast PBGS is 50%, at an average cost increase of 0.48% in the objective value across the three cases. None of the PHA solutions converges. The closest is when $\rho = 50000$, the violation reaches 0.06 before it triggers the stopping criterion. Note that the convergence tolerance ϵ for PHA is set to 0.01. The PHA simulation is terminated if no progress in violation is made for five consecutive iterations. Additionally, a terminated PHA simulation requires rounding-off of the fractional implementable obtained, followed by UC to ensure its primal feasibility.

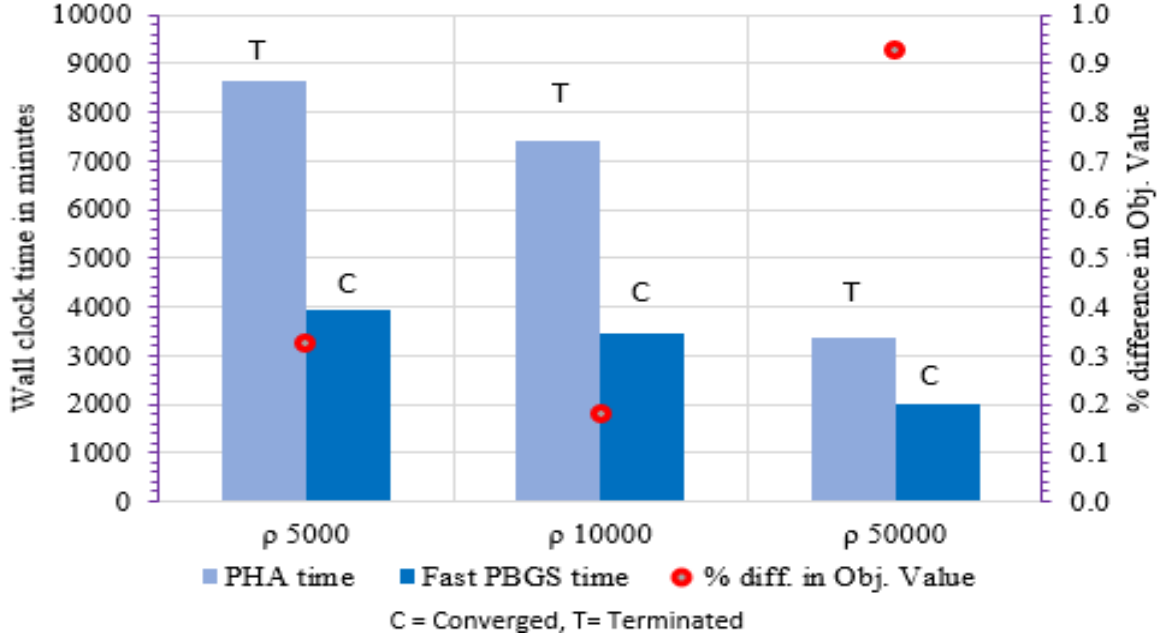


Figure 5.2 Comparison of Fast PBGS with PHA for the ERCOT-like System

In contrast, these additional heuristics and steps are not needed at all for the Fast PBGS since its convergence is guaranteed and the solution obtained is directly feasible. It is worth mentioning that in the case study of this research no cyclic behaviors of the PHA were observed as reported in [105]. Therefore, the performance of the PHA may be illustrated on the high side if potential cyclic behaviors are considered. Furthermore, some heuristic enhancements, such as using a large MIP gap for initial iterations, initializing successive iterations from the previous solution, and using cost proportional penalty factors, can be applied to both the Fast PBGS and the PHA to further improve their computational efficiency [105]. The results in Figure 5.2 show that the significant improvement in computational efficiency also makes the Fast PBGS a better choice on the practically-sized large-scale system.

5.5. Frank-Wolfe combined PHA (FW-PHA)

5.5.1. FW-PHA Formulation

If Step 10 step in Algorithm 3 can be modified to solve the problem over a convex hull $Conv(\Lambda^s)$ instead of Λ^s , then the PHA works for the SCUC problem and the dual solution converges to the optimal Lagrangian dual. However, the $Conv(\Lambda^s)$ is not readily available. Boland

et.al. have used simplicial decomposition to create a convex hull which is used in finding the lower bound [56]. This FW-PHA was applied to find lower bound for S-NCUC in [57]. The simplicial decomposition [106] is a generalization of the Frank-Wolf algorithm [107] in higher dimensions.

The convex hull created for each scenario (V_s) consists of the decision variables of linearized problem Θ^{+s} . That is minimizing the gradient approximation to Θ^{+s} at the point $(I_{i,t}^{s,k-1}, p_{d,i,t}^{s,k-1}, IU_{i,t}^s, ID_{i,t}^s)$.

$$\begin{aligned} \nabla \Theta^{+s}(\mathbf{I}^s, \mathbf{p}^s, \boldsymbol{\omega}^s, \boldsymbol{\rho}^s) \big|_{(I_{i,t}, p_{d,i,t}, IU_{i,t}, ID_{i,t}) = (I_{i,t}^{s,k-1}, p_{d,i,t}^{s,k-1}, IU_{i,t}^{s,k-1}, ID_{i,t}^{s,k-1})} \\ = \begin{bmatrix} NL_i + \omega_{i,t}^s + \rho(I_{i,t}^s - Z_{i,t}) \\ IC_{d,i} \\ SU_{i,t} \\ SD_{i,t} \end{bmatrix} = \begin{bmatrix} NL_i + \hat{\omega}_{i,t}^s \\ IC_{d,i} \\ SU_{i,t} \\ SD_{i,t} \end{bmatrix} \end{aligned} \quad (5.9)$$

solved for $\forall s \in S, \forall t \in NT, \forall i \in NI$ and $\forall d \in NG$. The resulting linearized problem is written as follows and solved as mixed integer linear program (MILP):

$$\begin{aligned} \hat{\Theta}^s(\hat{\mathbf{I}}^s, \hat{\mathbf{p}}^s, \hat{\mathbf{IU}}^s, \hat{\mathbf{ID}}^s) \\ = \sum_{t=1}^{NT} \left[\sum_{i=1}^{NI} \left((NL_i + \hat{\omega}_{i,t}^s) \cdot I_{i,t}^s + SU_{i,t} \cdot IU_{i,t}^s + \sum_{d=1}^{NG} (p_{d,i,t}^s \cdot IC_{d,i}) \right) \right. \\ \left. + (AL_t^s + LL_t^s) \cdot VOLL + \sum_{l \in L} (BrSl1_{l,t}^s + BrSl2_{l,t}^s) \cdot VOOB \right] \end{aligned} \quad (5.10)$$

Solution of this linearized problem $\widehat{\Theta}^s$ is used in creating elements of set V_s

- Unit Startup (\widehat{IU}^s)
- Unit Shutdown (\widehat{ID}^s)
- Unit Status (\widehat{I}^s)
- Block Dispatch (\widehat{p}^s).

$$V_s = V_s \cup \{ (\widehat{I}^s, \widehat{p}^s, \widehat{IU}^s, \widehat{ID}^s) \} \quad (5.11)$$

Solving the second problem as QP to obtain (I^s, p^s, IU^s, ID^s) as follows:

$$\{\Theta^{+s}(I^s, p^s, \omega^s, \rho^s) \mid (I^s, p^s, IU^s, ID^s) \in \text{conv}(V_s)\} \quad (5.12)$$

Note the constraint in the above problem. Instead of solving in the feasible region of Λ^s , this method solves in the convex hull of Λ^s . Problem (5.12) is accomplished by expressing (I^s, p^s, IU^s, ID^s) as a convex combination of the finite set of points, V_s , where the weights $a \in \mathbb{R}^{|V_s|}$ in the convex combination are now decision variables too. In fact, it is the only decision variable. In practice this can be implemented as follows:

$$(I^s, p^s, IU^s, ID^s) \in \underset{I, p, a}{\text{argmin}} \left\{ \begin{array}{l} \Theta^{+s}(I^s, p^s, \omega^s, \rho^s) \mid \\ (I^s, p^s, IU^s, ID^s) = \sum_{(\widehat{I}^{s,m}, \widehat{p}^{s,m}, \widehat{IU}^{s,m}, \widehat{ID}^{s,m}) \in V_s} a(\widehat{I}^{s,m}, \widehat{p}^{s,m}, \widehat{IU}^{s,m}, \widehat{ID}^{s,m}) , \\ \sum_{m=1}^{|V_s|} a_m = 1 \text{ and } a_m \geq 0, \forall m \end{array} \right\} \quad (5.13)$$

Presenting all the discussions in this section on FW-PHA in a flowchart format would make things more transparent. The flowchart is shown below in Figure 5.3

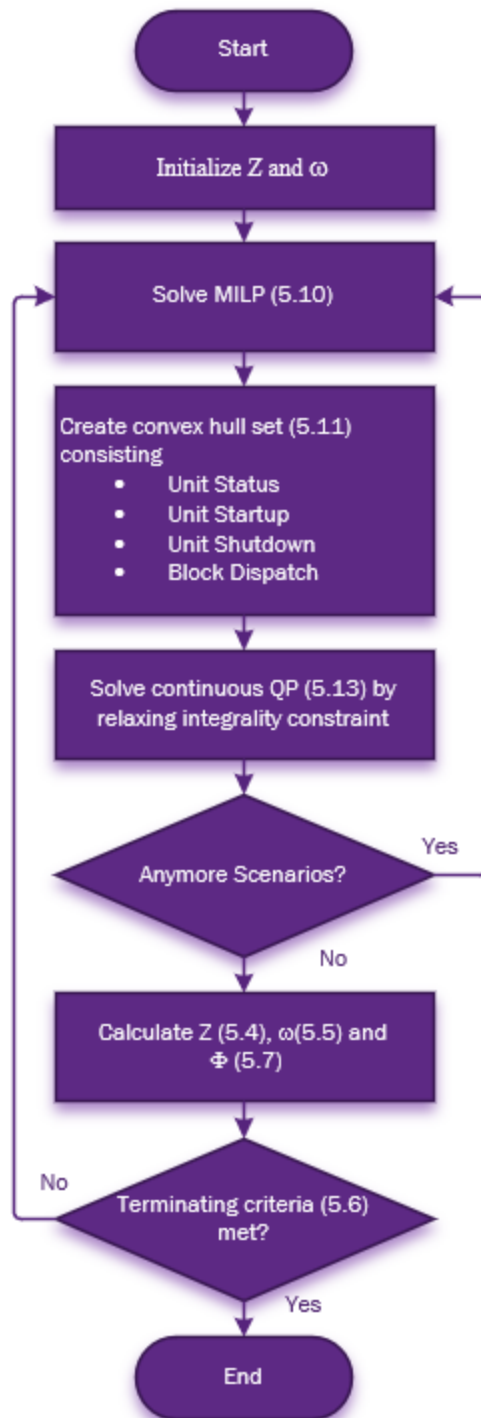


Figure 5.3 Flowchart of combined Frank-Wolfe and PHA

5.5.2. FW-PHA Initialization

Initialization of the convex set is important since the proposed FW-PHA method performs only one iteration of simplicial decomposition for each outer iteration. This requires that the initial scenario vertex sets share a common point.

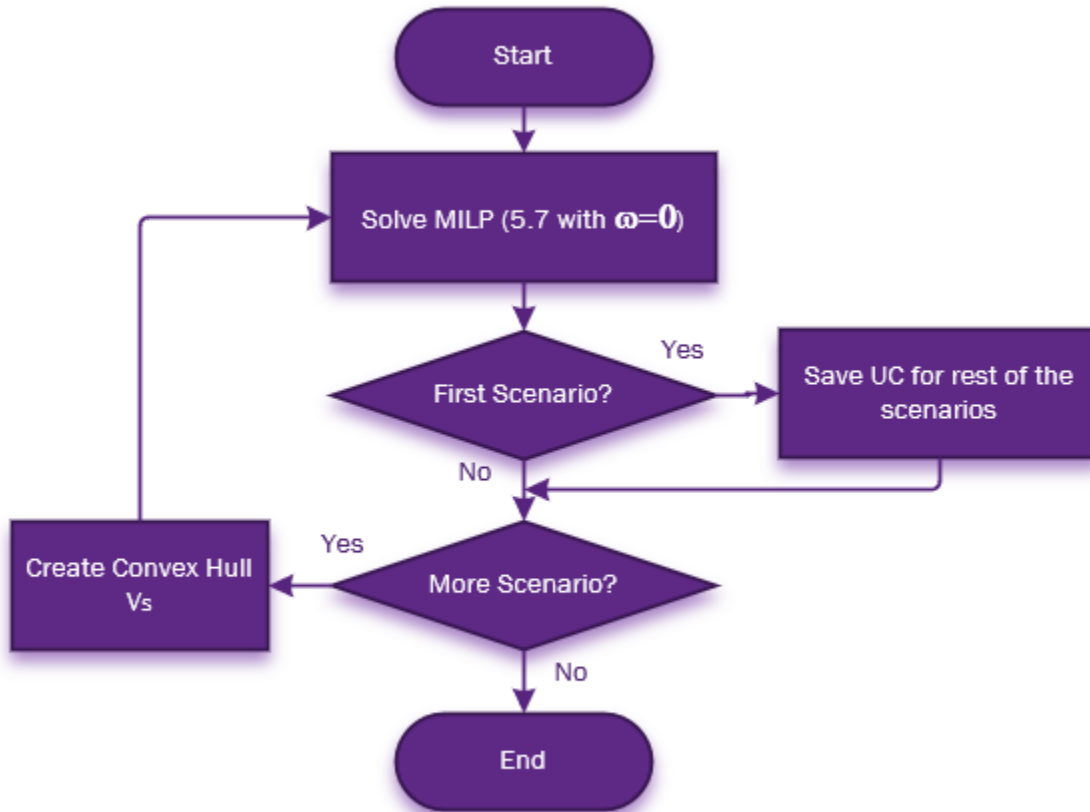


Figure 5.4 Initialization of convex hull for FW-PHA

Figure 5.4 shows the initialization of the convex hull for the FW-PHA algorithm. First, problem (5.7) is solved ($\omega = 0$) to obtain both first-stage and second-stage solutions for the first scenario. Then the rest of the scenarios are solved to get the second-stage decisions by fixing the first-stage unit commitment from the first scenario. The first-stage and second-stage solutions are used to initialize the inner approximation V_s , as shown in Figure 5.4. Both the implementable and the dual variables are calculated based on the above initialization.

5.5.3. FW-PHA with Warm Start

In the discussion on FW-PHA initialization, it was mentioned that the initialization of all scenarios must share the same vertex to use one inner iteration only. This requirement can be fulfilled by using a converged PBGS solution. The converged PBGS solution is the one where the implementable is the same for all scenarios. This can save a significant amount of time and allows to obtain better lower bound within the first few iterations.

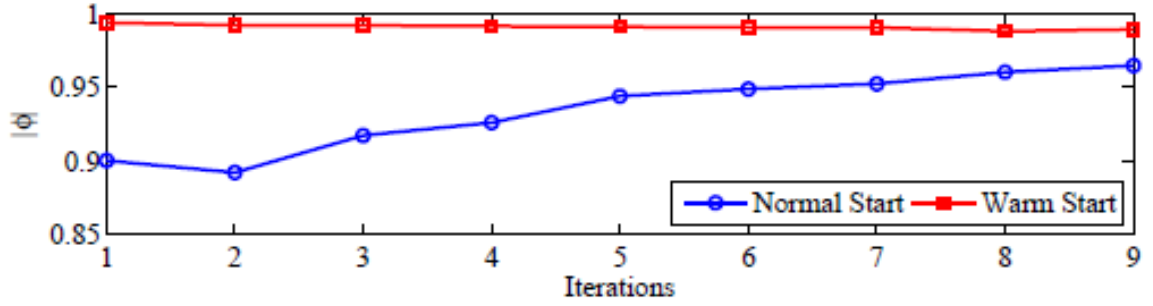


Figure 5.5 Lower bound using normal start vs warm-start for the IEEE 118-bus System

Figure 5.5 compares (normalized) FW-PHA lower bounds over iterations with normal and warm-starts on the IEEE-118 bus System. As seen, the normal-start lower bound improves as the iteration proceeds, whereas its warm-start counterpart is tighter and remain nearly unchanged over iterations. To save computational efforts, one can take the warm-start solution within the first few FW-PHA iterations as a lower bound on the PBGS solutions.

5.5.4. Comparison FW-PHA with PHA Lower Bound

FW-PHA was applied to determine lower bound for RTS-96 and the IEEE 118-bus Systems [57]. The results presented in this section are compared with lower bound obtained using PHA, which was explained in section 5.4 of this chapter.

5.5.4.1. RTS-96 System

The FW-PHA and PHA results for the set RTS96-10-S1 are show in the following figures. The major advantage of FW-PHA over PHA is sensitivity of the lower bound to the penalty factor ρ . This is evident in the Figure 5.6, where the increased ρ yields poor lower-bound. As the penalty

factor increases from 100 to 5,000, the lower bound obtained through the PHA dropped about 16.6% from \$331,350 to \$276,336. For the same range of penalty factors, the lower bound in the FW-PHA remains approximately the same (only 0.84% change).

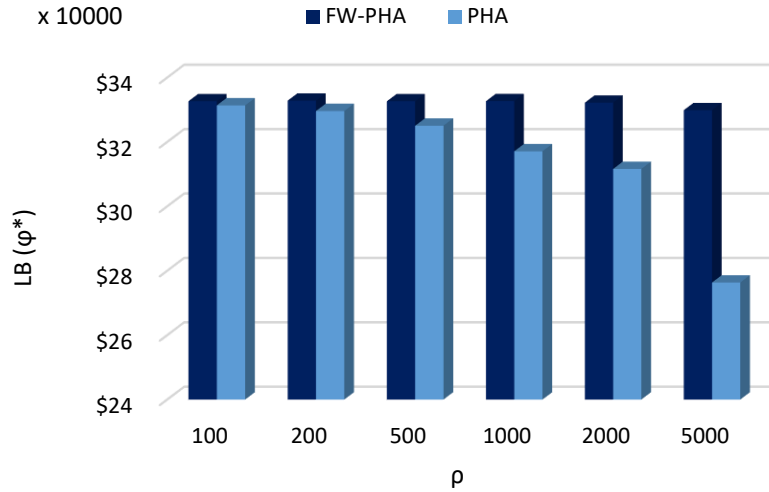


Figure 5.6 Sensitivity of lower bound to ρ for RTS-96 System

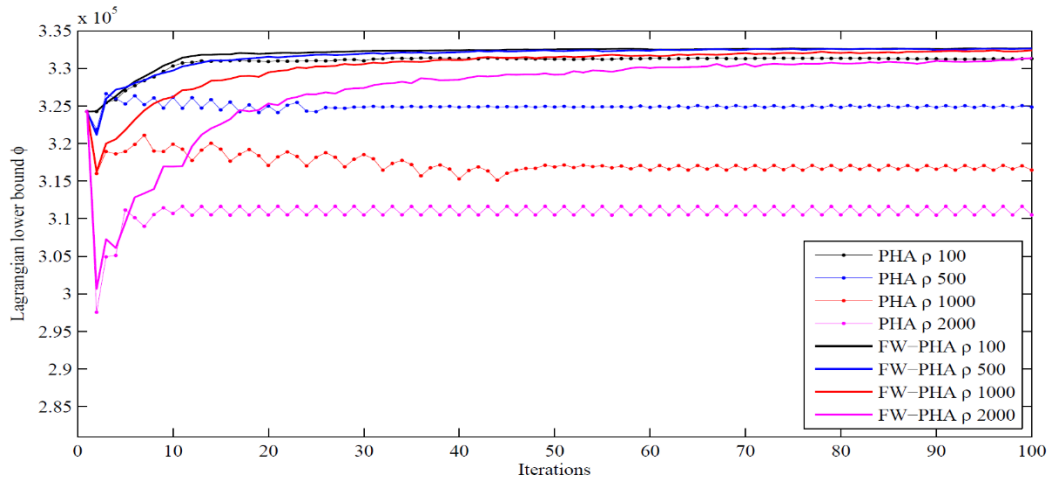


Figure 5.7 Lower bound over iterations for RTS-96 System

Figure 5.7 shows the lower bound obtained over iteration using the PHA (dotted plot) and the FW-PHA (line plot). Dotted lines obtained from the PHA can be distinctively seen, while solid lines from the FW-PHA are close to each other at the top of this figure. The lower bound from the PHA tends to settle at different values depending on the chosen ρ whereas the lower bound from

the FW-PHA, regardless of ρ , settles near the same value. Note that Figure 4 shows the cyclic behavior for the PHA with $\rho=1000$ (red-dotted) and $\rho=2000$ (magenta-dotted), which is an inherent drawback of the PHA mentioned earlier.

5.5.4.2. The IEEE 118-bus System

Figure 5.8 compares the sensitivity of the lower bound to the penalty factors in both the PHA and the FW-PHA. The lower bound obtained in the IEEE 118-bus System demonstrates a similar sensitivity to that in the IEEE RTS-96 System. The percentage drop of lower bound for the penalty factor increasing from 100 to 5,000 is about 6.6% in the PHA, while it is negligibly small, i.e., 0.3% in the FW-PHA.

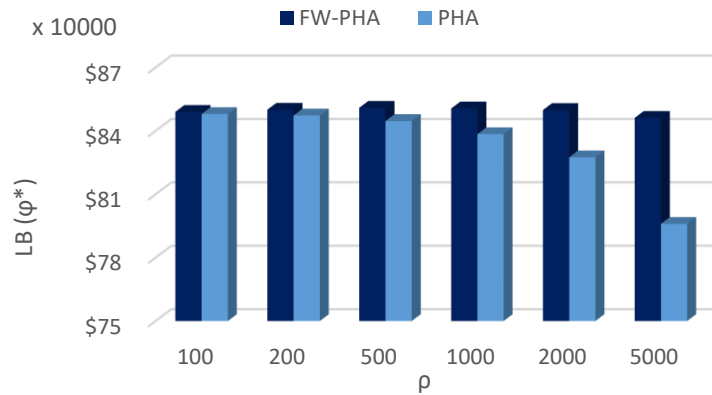


Figure 5.8 Sensitivity of lower bound to ρ for RTS-96 System

The lower bounds over iterations using the PHA (dotted plot) and the FW-PHA (line plot) are shown in Figure 5.9. This figure illustrates a similar trend to that in the modified IEEE RTS-96 System. The lower bound from the FW-PHA converges to around the same value regardless of ρ , while the lower bound in the PHA tends to settle at different values depending on ρ . It is worth mentioning that for $\rho=2,000$ in the PHA, this is the only case when the solution met our strict termination criteria, i.e., $\epsilon = 0.001$, while all other cases in Figure 5.9 were terminated at the maximum iterations.

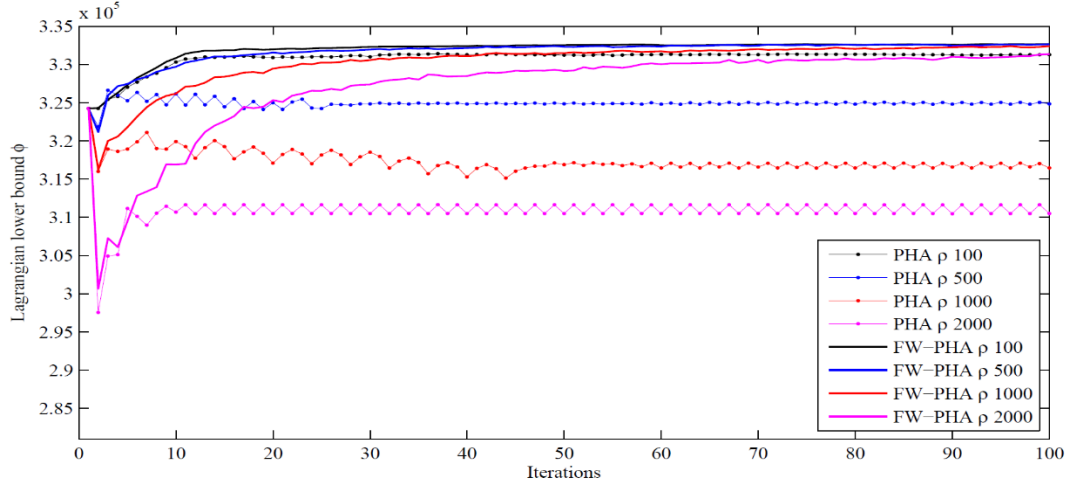


Figure 5.9 Lower bound over iterations for the IEEE 118-bus System (IEEE118-10-S1)

To illustrate the effectiveness of the proposed algorithm in a larger S-SCUC problem, a similar comparative case study is conducted on the IEEE 118-bus System with 50 scenarios under the same penalty factors as tested before. The lower bound versus iterations is shown in Figure 5.10. The convergence characteristics are analogous to those observed with 10 scenarios.

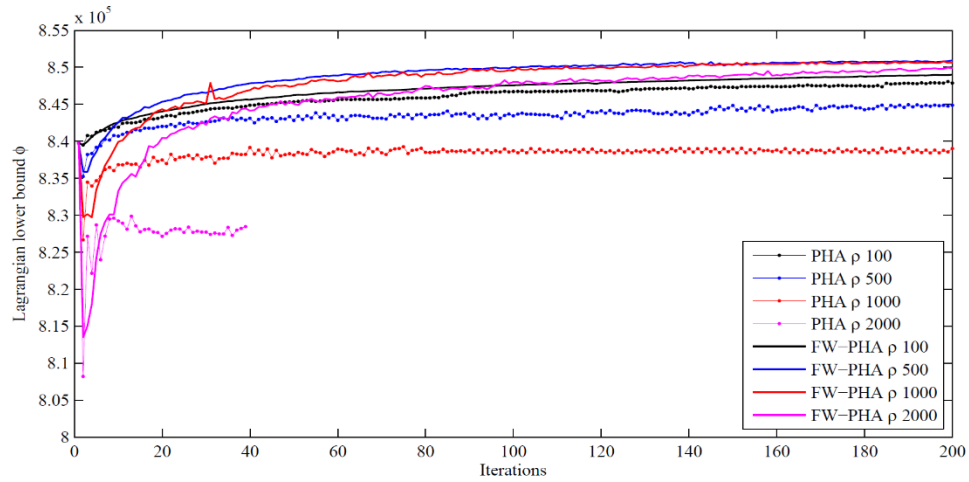


Figure 5.10 Lower bound over iterations for the IEEE 118-bus System (IEEE118-50-S0)

Through RTS-96 and the IEEE 118-bus Systems it has been demonstrated that FW-PHA lower bound is superior to that of PHA lower bound.

5.6. Evaluation of Fast PBGS results

In this section, appraisal of the Fast PBGS result with the FW-PHA lower bound method is carried out. Like before, all three different systems are assessed in this section.

5.6.1. RTS-96 System

The Fast PBGS results of the set RTS96-10-S1 are shown in Table 5.3. The lower bound obtained using FW-PHA is used in assessing the Fast PBGS quality. As can be seen, the Gaps for the Fast PBGS results obtained are slightly more than 1% for all penalty factors except 10,000. The actual gap of Fast PBGS is no higher than the one calculated using FW-PHA. It could be less.

Table 5.3 Assessment of Fast PBGS result using FW-PHA LB for RTS-96 System

$\gamma=1.0, \beta=1.1$		FW-PHA= \$330,598
$\underline{\rho}, \bar{\rho}$	Operating cost by Fast PBGS	Gap w.r.t. FW-PHA
500	\$330,990	1.03%
1000	\$334,313	1.12%
5,000	\$334,362	1.14%
10,000	\$337,156	1.98%

5.6.2. The IEEE 118-bus System

Table 5.4 shows the Fast PBGS results for the IEEE 118-bus System, set IEEE118-10-S1. Similar observation to RTS-96 in Table 5.3 can be made here. The calculated gap is around 1.5% and the actual gap could be less. Note that the FW-PHA lower bound result was obtained only few iterations into the solution.

Table 5.4 Assessment of Fast PBGS result using FW-PHA LB for the IEEE118-bus System

$\gamma=1.0, \beta=1.1$		FW-PHA= \$839,840
$\underline{\rho}, \bar{\rho}$	Operating cost by Fast PBGS	Gap w.r.t. FW-PHA
500	\$851,149	1.35%
1000	\$851,097	1.34%
5,000	\$851,366	1.37%
10,000	\$854,284	1.72%

5.6.3. The ERCOT-Like System

Table 5.5 shows the Fast PBGS results for the sizeable ERCOT-like System along with the gap calculated using FW-PHA lower bound. This gap calculated for the penalty factors 5,000 and 10,000 show that the Fast PBGS solution obtained is close to perfect.

Table 5.5 Assessment of Fast PBGS result using FW-PHA LB for the ERCOT-like System

$\gamma=1.0, \beta=1.1$		FW-PHA = \$35,356,211
$\underline{\rho}, \bar{\rho}$	Operating cost by Fast PBGS	Gap w.r.t. FW-PHA
5,000	\$35,444,677	0.25%
10,000	\$35,480,631	0.35%
50,000	\$35,880,952	1.48%

Through three different systems, each with different penalty factors, it can be established that the FW-PHA can be used as an effective lower bound method to evaluate the Fast PBGS solution.

5.7. Out-of-Sample Testing

For the current research, 25 sets (IEEE118-50-S1 through IEEE118-50-S25) of 50 scenarios were created and solved by EF, Fast PBGS, and PHA. The ARMA was used with 4% STD for load and 8% STD for wind in creating these scenarios. The Fast PBGS simulations used an initialization method for the implementable is different from the previous simulations. This initialization method is discussed in section 4.5.2. For PHA, no heuristic-based improvements suggested in [105] were used. The Fast PBGS can use some of those techniques to improve computational performance as well. While all the Fast PBGS converged to zero violations, only 10 out of 25 sets converged under PHA. The PHA termination criteria was set a little higher at $\epsilon \leq 0.1$ or 60 iterations whichever comes first. Table 5.6 shows production cost obtained by the EF, PHA and Fast PBGS. The objective value differences between the PHA and Fast PBGS are calculated with respect to the EF. The table includes, statistical information at the bottom.

Table 5.6 Comparison of Fast PBGS objective value with EF and PHA for the IEEE 118-bus System

Scenario Sets	Objective Value				
	EF	Fast PBGS		PHA	
	Prod. Cost	Prod. Cost	% diff w.r.t. EF	Prod. Cost	% diff w.r.t. EF
Set 1	\$ 905,258	\$ 908,365	0.34%	\$ 912,276	0.78%
Set 2	\$ 898,905	\$ 905,710	0.76%	\$ 921,790	2.55%
Set 3	\$ 878,844	\$ 885,622	0.77%	\$ 883,152	0.49%
Set 4	\$ 860,010	\$ 863,561	0.41%	\$ 862,267	0.26%
Set 5	\$ 886,673	\$ 893,280	0.75%	\$ 926,017	4.44%
Set 6	\$ 875,683	\$ 881,781	0.70%	\$ 889,085	1.53%
Set 7	\$ 904,196	\$ 912,470	0.91%	\$ 929,102	2.75%
Set 8	\$ 888,268	\$ 894,368	0.69%	\$ 967,259	8.89%
Set 9	\$ 870,647	\$ 873,525	0.33%	\$ 875,809	0.59%
Set 10	\$ 861,534	\$ 863,873	0.27%	\$ 861,999	0.05%
Set 11	\$ 933,354	\$ 958,349	2.68%	\$ 967,790	3.69%
Set 12	\$ 885,667	\$ 888,968	0.37%	\$ 917,521	3.60%
Set 13	\$ 856,417	\$ 861,333	0.57%	\$ 864,391	0.93%
Set 14	\$ 871,254	\$ 884,058	1.47%	\$ 873,579	0.27%
Set 15	\$ 908,052	\$ 911,445	0.37%	\$ 930,438	2.47%
Set 16	\$ 892,096	\$ 895,469	0.38%	\$ 893,505	0.16%
Set 17	\$ 872,411	\$ 895,985	2.70%	\$ 921,647	5.64%
Set 18	\$ 883,369	\$ 888,321	0.56%	\$ 906,165	2.58%
Set 19	\$ 903,864	\$ 906,125	0.25%	\$ 906,628	0.31%
Set 20	\$ 928,003	\$ 936,288	0.89%	\$ 935,542	0.81%
Set 21	\$ 878,289	\$ 885,465	0.82%	\$ 907,617	3.34%
Set 22	\$ 865,604	\$ 874,310	1.01%	\$ 887,009	2.47%
Set 23	\$ 911,826	\$ 936,660	2.72%	\$ 943,783	3.50%
Set 24	\$ 891,609	\$ 900,391	0.98%	\$ 901,054	1.06%
Set 25	\$ 874,521	\$ 885,005	1.20%	\$ 885,554	1.26%
STD	\$ 20,339	\$ 23,367	0.74%	\$ 29,836	2.07%
Min	\$ 856,417	\$ 861,333	0.25%	\$ 861,999	0.05%
Max	\$ 933,354	\$ 958,349	2.72%	\$ 967,790	8.89%
Mean	\$ 887,454	\$ 895,629	0.92%	\$ 906,839	2.18%
Median	\$ 885,667	\$ 893,280	0.75%	\$ 906,628	1.53%

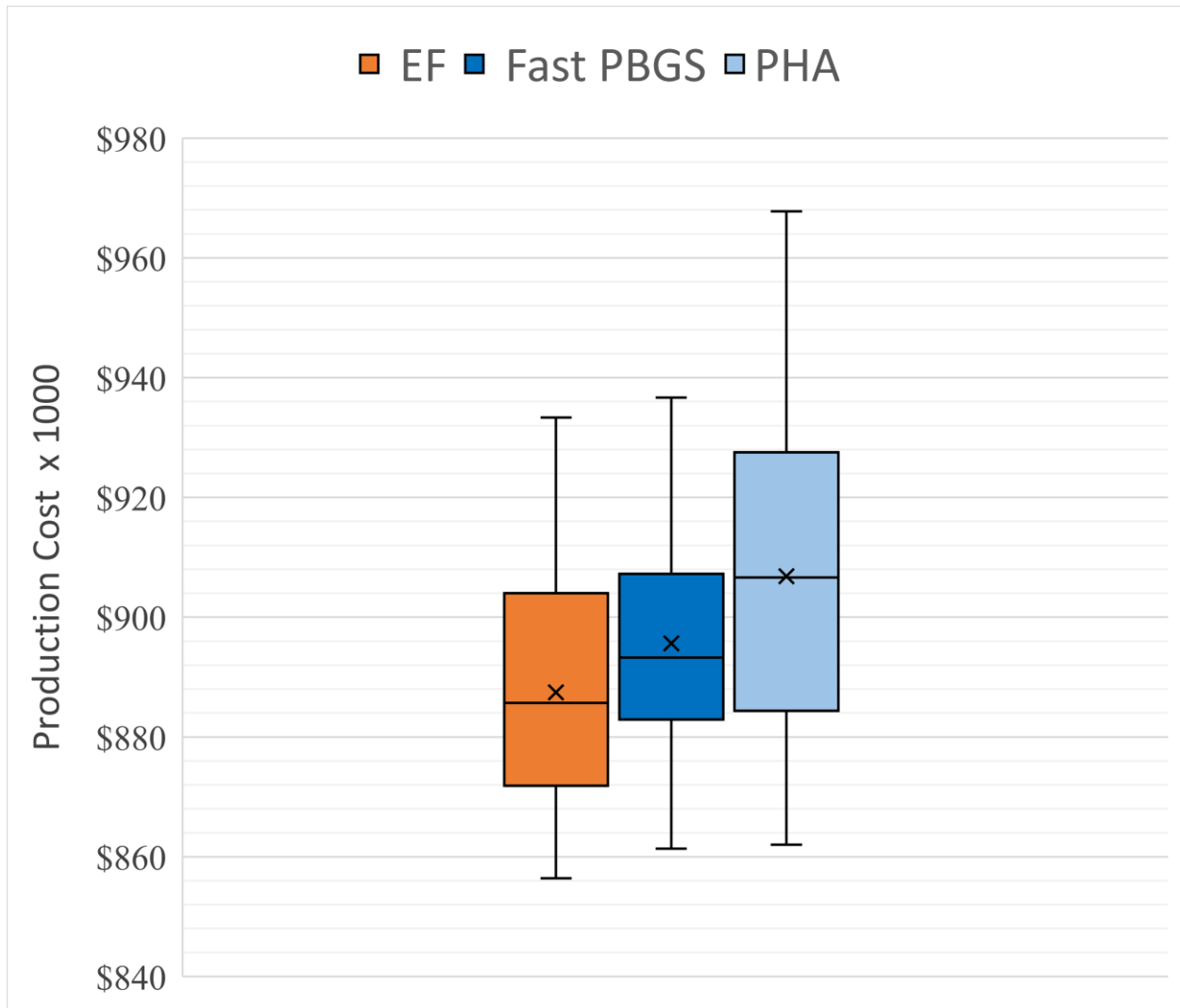


Figure 5.11 Comparison of objective values for the IEEE 118-bus system

Figure 5.11 shows the objective values of the three methods using the box-whiskers plot. As can be seen, the range of the objective values obtained by the Fast PBGS is almost the same as the range of the objective values obtained by the EF. There is an outlier that resulted in over 2% difference with EF. The mean difference is 0.92% with a standard deviation of 0.74% when compared with EF. The ‘x’ marker next to the median indicates the mean value. Both the median and mean difference of PHA are much higher than those of EF. The minimum value of PHA is almost at the 1st quartile of EF. Most of the 4th quartile of the PHA objective value is above the maximum value of the objective value obtained by EF. The average PHA difference with EF is 2.18% with a standard deviation of 2.07%. Overall, the PHA objective values are higher than both EF and the Fast PBGS. Most of the higher objective values could be attributed to the non-

convergence of PHA. Table 5.7 shows the computational time for EF, Fast PBGS and PHA. The comparison of the computational time for the Fast PBGS and PHA are made with EF. Statistical information is included at the bottom of the table. As can be seen the Fast PBGS outperforms both EF and the PHA.

Table 5.7 Comparison of Fast PBGS computational time with EF and PHA for the IEEE 118-bus System

Scenario Sets	Computational Time				
	EF	Fast PBGS		PHA	
	Time in min	Time in min	% diff w.r.t. EF	Time in min	% diff w.r.t. EF
Set 1	277	67	-76%	672	142%
Set 2	299	139	-54%	649	117%
Set 3	372	139	-63%	689	85%
Set 4	439	157	-64%	703	60%
Set 5	264	38	-86%	757	187%
Set 6	222	103	-53%	674	204%
Set 7	183	84	-54%	804	339%
Set 8	296	134	-55%	747	152%
Set 9	459	82	-82%	493	7%
Set 10	185	146	-21%	313	70%
Set 11	294	208	-29%	776	164%
Set 12	156	163	4%	711	355%
Set 13	192	144	-25%	612	219%
Set 14	175	286	63%	412	136%
Set 15	247	98	-60%	721	193%
Set 16	247	61	-75%	707	186%
Set 17	377	209	-45%	743	97%
Set 18	428	79	-82%	669	56%
Set 19	188	154	-18%	464	147%
Set 20	341	67	-80%	632	85%
Set 21	332	161	-51%	685	106%
Set 22	506	167	-67%	825	63%
Set 23	287	228	-21%	778	171%
Set 24	491	146	-70%	631	29%
Set 25	225	123	-45%	660	193%
STD	104	58	33.13%	123	84.54%
Min	156	38	-85.52%	313	7.48%
Max	506	286	63.47%	825	355.01%
Mean	299	135	-48.32%	661	142.55%
Median	287	139	-53.97%	685	142.25%

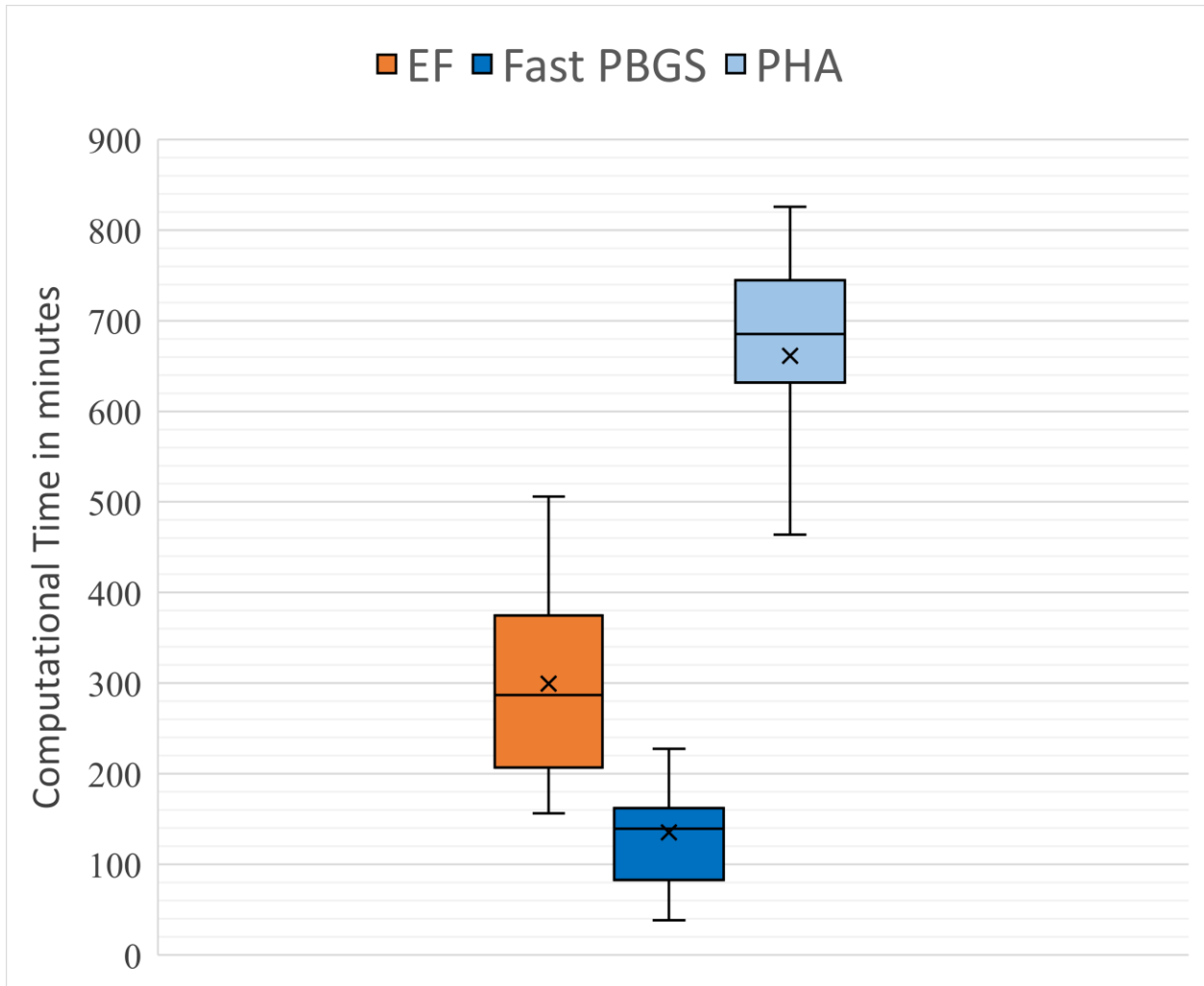


Figure 5.12 Comparison of computational performance for the IEEE 118-bus system

Figure 5.12 shows a comparison of the computational performance of the three methods. It is known that the computation time taken by the EF increases with an increased number of scenarios leading to the unsolvable situation. As shown in the figure, the Fast PBGS computational performance is far better than the PHA by several folds. The mean time taken to obtain the solution by the Fast PBGS is 135 minutes with a standard deviation of 58 minutes. Meanwhile, for the PHA algorithm, the mean is 661 minutes at a standard deviation of 123 minutes. Such a computational efficiency and comparable lower objective values shown here make the Fast PBGS a worthy alternative to the PHA.

5.8. Discussion on Parameters

Fast PBGS uses the parameters β , ρ and γ . Though the research continues on the sensitivity of these parameters, the experience thus far is shared here. The parameter β is used in (4.12) to accelerate the convergence. The research was experimented only with two values, 1.11 and 1.25. Results of the effect of β is shown in Figure 5.1. Simulation results for the IEEE 118-bus System show a high value of β makes the convergence faster at the cost of the solution quality. As the iteration number gets high, β grows exponentially and makes the penalty function higher, which could result in loss of load or additional load slack. The situation is not actual and can be fixed by increasing the Value of Loss of Load (VOLL) to obtain a solution or increasing the value of α in (4.12) only when needed instead of in every iteration.

Simulations were carried out in the current research with a wide range of ρ for both the IEEE 118-bus System and the ERCOT-like System. Like β , a high value of ρ makes the convergence faster at the cost of the solution quality. While ρ is used in the initial penalty, the parameter γ (step size) is used in every iteration (4.9). Though the values assigned to both ρ and γ were the same for the results shown in this dissertation, few simulations were carried out with different ρ and γ values. While large ρ with smaller γ took more iterations to make any significant progress, smaller ρ value compared to γ did not make that much difference. The progress remained when setting both ρ and γ at the same value. More studies need to be conducted before any conclusive statements are made and will be taken up in the future. The suggested range for ρ is 5,000 to 50,000. For the step size γ is set equal to ρ . The acceleration factor β can be set to 1.11 to 1.15.

One other experiment was carried out in initializing the \mathbf{Z} value. The algorithm in [101] suggests initializing $\mathbf{Z} = \mathbf{0}$ or using a rounded scenario weighted average of every status across the scenarios of the respective unit and hours. A new method as explained in Section 4.5.2 was used. This third method (4.34) initializes \mathbf{Z} with the scenario containing a maximum number of online unit status. Thus far, with the experiments in the present research, it was observed that this method of initialization gives results faster than the first two methods.

5.9. Summary

This chapter presents several methods that could be applied to measure the quality of the Fast PBGS solution. The methods included primal methods as well as lower-bound algorithms. The importance of measuring the solution quality of Fast PBGS was discussed as the algorithm does not use bounds to achieve convergence like Benders Decomposition, and applying the Fast PBGS to MIP such as S-NCUC often gives local minimums than the global one. Of those lower-bound methods presented here, it was shown that the FW-PHA method is the one that gives the best lower bound for real-world applications. The FW-PHA solution time has been improved through warm starting with a converged Fast PBGS solution. Also, a comparison of Fast PBGS with PHA was made as PHA is one of the widely used scenario-wise decompositions. Finally, out-of-sampling (OOS) tests were conducted on the IEEE 118-bus system. A total of 25 sets, each with 50 scenarios, were used in the OOS tests. Three different algorithms, EF, PHA, and Fast PBGS, were measured and compared. Comparisons on the quality of the objective value and the computational time were carried out and quantified. The comparative results with respect to EF (i.e., optimal solution) shows the Fast PBGS results are closer to that of the EF with an average difference of 0.92%, whereas the PHA solution with respect to the EF with an average difference of 2.18%. When it comes to the computational time, the Fast PBGS outperformed both EF and PHA. Specifically, the Fast PBGS took on average 48% of less time than EF to obtain a solution. Compared with PHA, the Fast PBGS was 142% faster.

6. Conclusions and Future Work

The generation landscape of the utilities and the ISOs is changing in favor of VRE. This change brings uncertainties into the PSOP. One of the applications used in the PSOP is the UC (NCUC or SCUC). What used to be parameters have uncertainties in them now, therefore, needing changes to the application. After a brief discussion on a different solution to this problem, it was decided to research in the stochastic optimization area to make the research applicable to real-world situations. First, we formulated the EF of the NCUC and conducted studies on three different systems with varying sizes. One of the systems included was the ERCOT-like Large System. We learned that we could not obtain a solution for this extensive complex system using EF. This prompted us to look into decomposition and found PHA and BD algorithms. Both these algorithms are already researched. BD algorithm would require significant effort to implement around an existing solution at the utilities and ISOs compared to the PHA. However, the PHA has drawbacks prompting the following question: 1) How can we devise an effective penalty function to obtain an exact solution with a zero-duality gap? and 2) If an exact solution is attained, how can we find a robust yet tight lower bound capable of measuring the quality of the exact solution accurately?

An answer was found for the first question using the PBGS algorithm. Though a solution was obtained to the ERCOT-like Large System, the time taken to obtain such a solution prohibits from using it in the real world. This challenge steered the research to find a solution in Fast PBGS. The Fast PBGS is an improved version of the PBGS and saves an average of 35%–50% of the time compared to the PBGS. An answer to the second question was found in FW-PHA lower bound algorithm. The use of the warm-start technique improved the computation performance of FW-PHA.

To prove the reliability of the Fast PBGS algorithm, OOS was conducted on the IEEE 118-bus system. A total of 25 sets, each with 50 scenarios, were generated and studied in the OOS using EF, PHA, and the Fast PBGS. A comparison of results obtained for these three algorithms shows that the Fast PBGS solution is very close to the EF than the PHA solution to the EF solution. In the performance comparison, the Fast PBGS outperforms both PHA and EF.

Though goals set out was achieved, more improvements, especially in the computational performance area, are needed. Future research can be conducted in the following areas to improve

computational performance. 1) Improved UC formulation: Most time taken is in obtaining an NCUC solution. Time taken can be reduced by using improvements made in the problem formulation or other solution techniques in the research, such as the one in [77]; 2) PHA expediting techniques: Computation performance can be improved by using expediting techniques used in PHA [105]; 3) An optimal value for tunable parameters: More studies can be carried out with different tunable parameters to arrive at optimal values of these parameters that improve the performance; and 4) Parallel implementation of the Fast PBGS algorithm: The decomposition lets one solve the scenarios in parallel. Perhaps using the latest advancement in parallel implementation such as the one in [108]. In addition, machine learning can also be explored to obtain a solution without actually solving it. This remains an interesting topic deserving full investigation for other researchers in the future.

References

- [1] Leon K. Kirchmayer. Economic Operation of Power Systems. John Wiley & Sons, New York, 1958.
- [2] J.A. Muckstadt, R.C. Wilson, “An Application of Mixed-Integer Programming Duality to Scheduling Thermal Generating Systems”, *IEEE Trans. Power Apparatus Syst.*, vol. 87, issue 12, pp. 1958-1978, 1968.
- [3] ERCOT, “Generator interconnection status report”
(<http://www.ercot.com/gridinfo/resource/2017>) last accessed 07/15/2020.
- [4] Southwest Power Pool, “State of the market 2018”,
<https://spp.org/documents/59861/2018%20annual%20state%20of%20the%20market%20report.pdf>, last accessed 07/17/2020.
- [5] California ISO, “2019 annual report on market issues & performances”,
<http://www.caiso.com/Documents/2019AnnualReportonMarketIssuesandPerformance.pdf#search=generation%20fuel%20mix>, last accessed 07/17/2020.
- [6] Energy.Gov, “U.S. installed and potential wind power capacity and generation”,
<https://windexchange.energy.gov/maps-data/321>, last accessed 07/17/2020.
- [7] W. van Ackooij *et al.*, “Large-scale unit commitment under uncertainty: An updated literature survey,” *Annals of Operations Research.*, vol. 271, no. 1, pp. 11-85, 2018.
- [8] Q. P. Zheng, J. Wang, and A. L. Liu, “Stochastic optimization for unit commitment—A review,” *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1913–1924, 2015.
- [9] L. T. Anstine, *et.al.*, “Application of probability methods to the determination of spinning reserve requirements for the Pennsylvania-New Jersey-Maryland Interconnection,” *IEEE Trans. Power Apparatus Syst.*, vol. 82, no. 68, pp. 726–735, 1963.
- [10] ERCOT, “Methodology for determining ancillary service requirements”,
<http://www.ercot.com/mktinfo/dam/index.html>, last accessed 07/18/2020.

- [11] M. A. Matos and R. J. Bessa, "Setting the operating reserve using probabilistic wind power forecasts," *IEEE Trans. Power Syst.*, vol. 26, no. 2, pp. 594–603, 2011.
- [12] H. Holttinen *et al.*, "Methodologies to determine operating reserves due to increased wind power," *IEEE Trans. Sustain. Energy*, vol. 3, no. 4, pp. 713–723, 2012.
- [13] N. Menemenlis, M. Huneault, and A. Robitaille, "Computation of dynamic operating balancing reserve for wind power integration for the time-horizon 1–48 Hours," *IEEE Trans. Sustain. Energy*, vol. 3, no. 4, pp. 692–702, 2012.
- [14] K. De Vos, "Sizing and allocation of operating reserves following wind power integration," Ph.D. dissertation, KU Leuven, Belgium, April 2013.
- [15] K. D. Vos and J. Driesen, "Dynamic operating reserve strategies for wind power integration," *IET Renewable Power Generation*, vol. 8, no. 6, pp. 598–610, 2014.
- [16] E. Ela, M. Milligan, and B. Kirby, "Operating reserves and variable generation," NREL/TP-5500-51978, 1023095, Aug. 2011.
- [17] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Management Science*, vol. 6, no. 1, pp. 73–79, 1959.
- [18] U. A. Ozturk, M. Mazumdar, and B. A. Norman, "A solution to the stochastic unit commitment problem using chance constrained programming," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1589–1598, 2004.
- [19] J. J. Peralta, J. Pérez-Ruiz, and S. de la Torre, "Unit commitment with load uncertainty by joint chance-constrained programming," in *2013 IEEE Grenoble Conference*, pp. 1–6, 2013.
- [20] H. Wu, M. Shahidehpour, Z. Li, and W. Tian, "Chance-constrained day-ahead scheduling in stochastic power system operation," *IEEE Trans. Power Syst.*, vol. 29, no. 4, pp. 1583–1591, 2014.
- [21] U. A. Ozturk, "The stochastic unit commitment problem: a chance constrained programming approach considering extreme multivariate tail probabilities." Ph.D. Thesis, University of Pittsburgh, PA, 2003.

- [22] Y. Wang, Q. Xia, and C. Kang, "Unit commitment with volatile node injections by using interval optimization," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1705–1713, 2011.
- [23] M. Zhou, S. Xia, G. Li, and X. Han, "Interval optimization combined with point estimate method for stochastic security-constrained unit commitment," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 276–284, 2014.
- [24] Y. Yu *et al.*, "Transmission contingency-constrained unit commitment with high penetration of renewables via interval optimization," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1410–1421, 2017.
- [25] D. You *et al.*, "An interval unit commitment with wind power integrated using interval optimization," in *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pp. 701–705, 2019.
- [26] L. Wu, M. Shahidehpour, and Z. Li, "Comparison of scenario-based and interval optimization approaches to stochastic SCUC," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 913–921, 2012.
- [27] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations Research Letters*, vol. 25, no. 1, pp. 1–13, 1999.
- [28] A. Ben-Tal and A. Nemirovski, "Robust optimization – methodology and applications," *Math. Program.*, vol. 92, no. 3, pp. 453–480, 2002.
- [29] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Math. Program.*; vol. 99, no. 2, pp. 351–376, 2004.
- [30] B. L. Gorissen, I. Yanıkoğlu, and D. den Hertog, "A Practical guide to robust optimization," *Omega*, vol. 53, pp. 124–137, 2015.
- [31] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.

- [32] A. Street, F. Oliveira, and J. M. Arroyo, "Contingency-constrained unit commitment with n-K security criterion: A robust optimization approach," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1581–1590, 2011.
- [33] R. Jiang, J. Wang, and Y. Guan, "Robust unit commitment with wind power and pumped storage hydro," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 800–810, 2012.
- [34] C. Zhao, J. Wang, J.-P. Watson, and Y. Guan, "Multi-stage robust unit commitment considering wind and demand response uncertainties," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2708–2717, 2013.
- [35] Q. Wang, J.-P. Watson, and Y. Guan, "Two-stage robust optimization for N-k contingency-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2366–2375, 2013.
- [36] D. Bertsimas *et.al.*, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 52–63, 2013.
- [37] C. Ning and F. You, "Data-driven adaptive robust unit commitment under wind power uncertainty: A Bayesian nonparametric approach," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 2409–2418, 2019.
- [38] Y. An and B. Zeng, "Exploring the modeling capacity of two-stage robust optimization: variants of robust unit commitment model," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 109–122, 2015.
- [39] C. Zhao and Y. Guan, "Unified stochastic and robust unit commitment," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3353–3361, 2013.
- [40] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Computational Management Science; Dordrecht*, vol. 2, no. 1, pp. 3–19, 2005.
- [41] R. M. Van Slyke and R. Wets, "L-Shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics*, vol. 17, no. 4, pp. 638–663, 1969.

- [42] L. F. B. Baptistella and J. C. Geromel, “Decomposition approach to problem of unit commitment schedule for hydrothermal systems,” *IEE Proceedings D - Control Theory and Applications*, vol. 127, no. 6, pp. 250–258, 1980.
- [43] M. Shahidehpour and Y. Fu, “Tutorial Benders decomposition in restructured power systems,” <http://motor.ece.iit.edu/ms/benders.pdf> (last accessed 09/30/2020)
- [44] C. Liu, M. Shahidehpour, and L. Wu, “Extended Benders decomposition for two-stage SCUC,” *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 1192–1194, 2010
- [45] L. Wu and M. Shahidehpour, “Accelerating the Benders decomposition for network-constrained unit commitment problems,” *Energy Systems; Gainesville*, vol. 1, no. 3, pp. 339–376, 2010.
- [46] B. Vandenbussche, S. Delikaraoglou, I. Blanco, and G. Hug, “Data-driven adaptive Benders decomposition for the stochastic unit commitment problem,” Dec. 2019, <http://arxiv.org/abs/1912.01039> (last accessed 09/18/2020)
- [47] J. Murphy, “Benders, nested Benders and stochastic programming,” *Cambridge University Engineering Department Technical Report*, Dec 2013.
- [48] C. C. Carøe and R. Schultz, “Dual decomposition in stochastic integer programming,” *Operations Research Letters*, vol. 24, no. 1–2, pp. 37–45, 1999.
- [49] S. Takriti, J. R. Birge, and E. Long, “A stochastic model for the unit commitment problem,” *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1497–1508, 1996.
- [50] L. Wu, M. Shahidehpour, and T. Li, “Stochastic security-constrained unit commitment,” *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 800–811, 2007.
- [51] R. T. Rockafellar and R. J.-B. Wets, “Scenarios and policy aggregation in optimization under uncertainty,” *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119–147, 1991.
- [52] Løkketangen, A., Woodruff, D.L. “Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming,” *J Heuristics* 2, pp. 111–128, 1996.

- [53] Y. Fan and C. Liu, “Solving stochastic transportation network protection problems using the progressive hedging-based method,” *Netw Spat Econ*, vol. 10, no. 2, pp. 193–208, 2010.
- [54] J.-P. Watson and D. L. Woodruff, “Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems,” *Computational Management Science*, vol. 8, no. 4, pp. 355–370, 2011.
- [55] D. Gade *et.al.*, “Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs,” *Math. Program.*, vol. 157, no. 1, pp. 47–67, 2016.
- [56] N. Boland *et al.*, “Combining progressive hedging with a Frank-Wolfe method to compute Lagrangian dual bounds in stochastic mixed-integer programming,” *SIAM J. Optim.*, vol. 28, no. 2, pp. 1312–1336, 2018.
- [57] A. M. Palani, H. Wu, and M. M. Morcos, “A Frank–Wolfe progressive hedging algorithm for improved lower bounds in stochastic SCUC,” *IEEE Access*, vol. 7, pp. 99398–99406, 2019.
- [58] B. Colonetti and E. C. Finardi, “Combining Lagrangian relaxation, benders decomposition, and the level bundle method in the stochastic hydrothermal unit-commitment problem,” *International Transactions on Electrical Energy Systems*, vol. 30, issue 9, 2020.
- [59] C. Zhao and Y. Guan, “Unified stochastic and robust unit commitment,” *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3353–3361, 2013.
- [60] Wood, A. J., and Wollenberg, B. F. *Power Generation, Operation, and Control*. John Wiley and Sons, New York, NY, 1984.
- [61] M. Carrion and J. M. Arroyo, “A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem,” *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1371–1378, 2016.
- [62] H. Ma and S. M. Shahidehpour, “Unit commitment with transmission security and voltage constraints,” *IEEE Trans. Power Syst.*, vol. 14, no. 2, pp. 757–764, 1999.

- [63] V. Trovato, A. Bialecki, and A. Dallagi, “Unit commitment with inertia-dependent and multispeed allocation of frequency response services,” *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1537–1548, 2019.
- [64] D. A. Tejada-Arango, S. Lumbreras, P. Sánchez-Martín, and A. Ramos, “Which unit-commitment formulation is best? A comparison framework,” *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2926–2936, 2020.
- [65] N. P. Padhy, “Unit commitment-a bibliographical survey,” *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1196–1205, 2004.
- [66] B. Knueven, J. Ostrowski, and J.-P. Watson, “On mixed-integer programming formulations for the unit commitment problem,” *INFORMS Journal on Computing*, p. ijoc.2019.0944, 2020.
- [67] E. Ela and M. O’Malley, “Studying the variability and uncertainty impacts of variable generation at multiple timescales,” *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1324–1333, 2012.
- [68] E. Ela, “Scheduling and pricing of power systems with increased variable, uncertain, and nonsynchronous resources,” Ph.D. dissertation, Univ. College Dublin, Dublin, Ireland, 2014.
- [69] T. Ibaraki, “On the computational efficiency of branch-and-bound algorithms,” *JORSJ*, vol. 20, no. 1, pp. 16–35, 1977.
- [70] A. J. Conejo, Ed., *Decomposition techniques in mathematical programming: engineering and science applications*. Berlin ; New York: Springer, 2006.
- [71] GAMS Documentation. [Online]. Available: <https://www.gams.com/latest/docs/> (last accessed 06/18/2021).
- [72] IBM ILOG CPLEX. [Online]. Available: <https://www.ibm.com/products/ilog-cplex-optimization-studio/resources> (last accessed 06/18/2021).
- [73] C. Grigg *et al.*, “The IEEE Reliability Test System-1996. A report prepared by the reliability test system task force of the application of probability methods subcommittee,” *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 1010–1020, 1999.

- [74] “IEEE 96-RTS Test System - Illinois Center for a Smarter Electric Grid (ICSEG).” <https://icseg.iti.illinois.edu/power-cases/ieee-96-rts-test-system/> (last accessed 11/ 02/2019).
- [75] IEEE118bus_Data_Figure.xls. [Online]. Available: <http://motor.ece.iit.edu/Data/> (last accessed 01/03/2019.)
- [76] Q. Wang *et.al.*, “Quantifying the economic and grid reliability impacts of improved wind power forecasting,” *IEEE Trans. Sustain. Energy*, vol. 7, no. 4, pp. 1525–1537, 2016.
- [77] X. Li, Q. Zhai, J. Zhou, and X. Guan, “A variable reduction method for large-scale unit commitment,” *IEEE Trans. Power Syst.*, pp. 1–1, 2019.
- [78] J. M. Morales, S. Pineda, A. J. Conejo, and M. Carrion, “scenario reduction for futures market trading in electricity markets,” *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 878–888, 2009.
- [79] Y. Dvorkin, Y. Wang, H. Pandzic, and D. Kirschen, “Comparison of scenario reduction techniques for the stochastic unit commitment,” in *2014 IEEE PES General Meeting Conference Exposition*, pp. 1–5, 2014.
- [80] P. Pinson, G. Papaefthymiou, B. Klockl, and H. A. Nielsen, “Generation of statistical scenarios of short-term wind power production,” *Lausanne Power Tech*, Switzerland, pp. 491–496, 2007.
- [81] D. Lee and R. Baldick, “Load and wind power scenario generation through the generalized dynamic factor model,” *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 400–410, 2017.
- [82] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, “Model-free renewable scenario generation using generative adversarial networks,” *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, 2018.
- [83] R. Henrion, W. Römisch, “Problem-based optimal scenario generation and reduction in stochastic programming,” *Math. Program.*, pp. 1–23, 2018.
- [84] A. Boone, “Simulation of short-term wind speed forecast errors using a multi-variate ARMA (1,1) Time-series Model,” p. 95.

- [85] Y. Liu, R. Sioshansi, and A. J. Conejo, “Multistage stochastic investment planning with multiscale representation of uncertainties and decisions,” *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 781–791, 2018.
- [86] A. Nedich and A. Ozdaglar, “A geometric framework for nonconvex optimization duality using augmented Lagrangian functions,” *J Glob Optim*, vol. 40, no. 4, pp. 545–573, 2008.
- [87] R.T. Rockafellar, R.J.B Wets. Variational Analysis. 3rd printing 2009. Available at <http://sites.math.washington.edu/~rtr/papers/rtr169-VarAnalysis-RockWets.pdf>
- [88] C.Y. Wang, X.Q. Yang, X.M. Yang, “Nonlinear augmented Lagrangian and duality theory,” *Mathematics of Operations Research* 38(4), 740-760, 2013.
- [89] M. J. Feizollahi, S. Ahmed, and A. Sun, “Exact augmented Lagrangian duality for mixed integer linear programming,” *Math. Program.*, vol. 161, no. 1–2, pp. 365–387, 2017.
- [90] M. J. Feizollahi, “Large-scale unit commitment: Decentralized mixed integer programming approaches”, Ph.D. dissertation, Georgia Institute of Technology, Dec 2015.
- [91] J. V. Burke, “An exact penalization viewpoint of constrained optimization,” *SIAM Journal on Control and Optimization*, vol. 29, no. 4, pp. 968–998, 1991.
- [92] G. Di Pillo and L. Grippo, “Exact penalty functions in constrained optimization,” *SIAM Journal on Control and Optimization; Philadelphia*, vol. 27, no. 6, p. 28, 1989.
- [93] J. V. Burke, “An exact penalization viewpoint of constrained optimization,” *SIAM Journal on Control and Optimization; Philadelphia*, vol. 29, no. 4, p. 31, 1991.
- [94] I. D. Coope and C. J. Price, “A two-parameter exact penalty function for nonlinear programming,” *J Optim Theory Appl*, vol. 83, no. 1, pp. 49–61, 1994.
- [95] R. S. Burachik, A. N. Iusem, and J. G. Melo, “Duality and exact penalization for general augmented Lagrangians,” *J Optim Theory Appl*, vol. 147, no. 1, pp. 125–140, 2010.
- [96] X. X. Huang and X. Q. Yang, “A unified augmented Lagrangian approach to duality and exact penalization,” *Mathematics of Operations Research.*, 28, 533–552, 2003.

- [97] M. V. Dolgopoliuk, "A Unifying theory of exactness of linear penalty functions," *Optimization*, vol. 65, no. 6, pp. 1167–1202, 2016.
- [98] M. V. Dolgopoliuk, "Existence of augmented Lagrange multipliers: reduction to exact penalty functions and localization principle," *Math. Program.*, vol. 166, no. 1–2, pp. 297–326, 2017.
- [99] S. Lucidi and F. Rinaldi, "Exact penalty functions for nonlinear integer programming problems," *J Optim Theory Appl*, vol. 145, no. 3, pp. 479–488, 2010.
- [100] N. L. Boland and A. C. Eberhard, "On the augmented Lagrangian dual for integer programming," *Math. Program.*, vol. 150, no. 2, pp. 491–509, 2015.
- [101] F. Oliveira, J. Christiansen, B. Dandurand, and A. Eberhard, "Combining penalty-based and Gauss-Seidel methods for solving stochastic mixed-integer problems," Jan. 2017 <http://arxiv.org/abs/1702.00074> (last accessed 03/13/2019)
- [102] C. Beltran, C. Taddonji and J.-Ph. Vial, "Semi-Lagrangian relaxation," September 14, 2004. (http://www.optimizationonline.org/DB_FILE/2004/09/959.pdf). Last accessed 7/14/2019.
- [103] X. Guan, Q. Zhai, and A. Papalexopoulos, "Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Jul. 2003, pp. 1095–1100.
- [104] A. M. Palani, H. Wu and M. M. Morcos, "A fast penalty-based Gauss-Seidel method for stochastic unit commitment with uncertain load and wind generation," in *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 211–222, 2021.
- [105] K. Cheung *et al.*, "Toward scalable stochastic unit commitment: Part 2: solver configuration and performance assessment," *Energy Syst*, vol. 6, no. 3, pp. 417–438, 2015.
- [106] M. Frank, P. Wolfe, "An algorithm for quadratic programming," *Naval Res Logist Quart* 3, 95–110, 1956.
- [107] B. Von Hohenbalken, "Simplicial decomposition in nonlinear programming algorithms," *Math. Program.* 13, 49 - 68, 1977.

[108] Y. Song, C. Meng, R. Liao, and S. Ermon, “Accelerating Feedforward Computation via Parallel Nonlinear Equation Solving,” Jun. 2021, <http://arxiv.org/abs/2002.03629> (last accessed 07/27/ 2021)